1987

Application Of Digital Signal Analysis, Mass Data Acquisition and Processing Techniques, and Automated Experiment Protocols to the Study of Cardiac Cell Membrane Electrophysiology, with Mathematical Modeling

David S. Buckles
Medical University of South Carolina

APPLICATION OF DIGITAL SIGNAL ANALYSIS, MASS DATA ACQUISITION AND PROCESSING TECHNIQUES, AND AUTOMATED EXPERIMENT PROTOCOLS TO THE STUDY OF CARDIAC CELL MEMBRANE ELECTROPHYSIOLOGY, WITH MATHEMATICAL MODELING

by

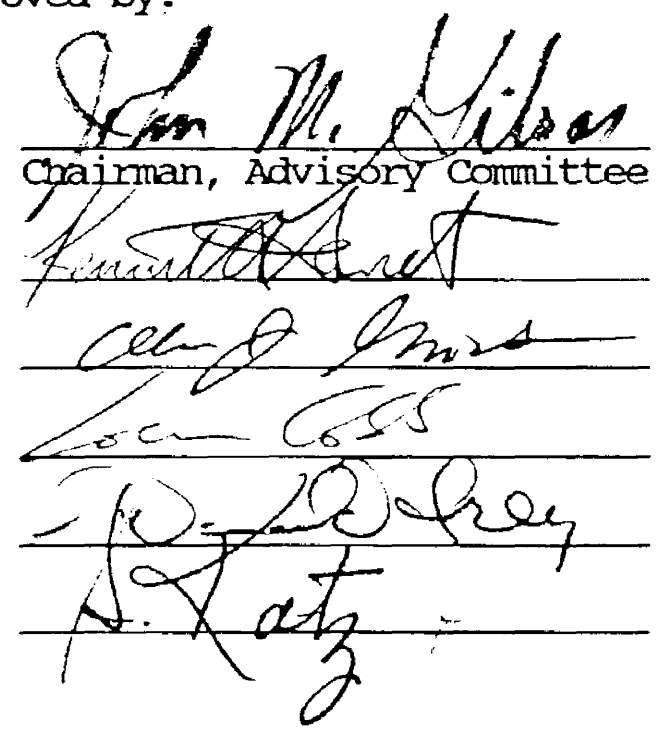
David S. Buckles

A dissertation submitted to the faculty of
the Medical University of South Carolina
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in the College of Graduate Studies

Department of Biometry

1987

Approved by:



Chairman, Advisory Committee

TABLE OF CONTENTS

DAVID S. BUCKLES. Digital Systems Science in Cardiac Electrophysiology. (Under the direction of Dr. John Gibson)

ABSTRACT

Traditional methods of collecting, analyzing and storing data from cardiac cell membrane electrophysiology experiments have become increasingly cumbersome and unwieldy as experimental protocols have become more sophisticated and complex. A global approach to collecting, analyzing, refining and storing electrophysiologic data, as well as a new approach to mathematical modeling of cell membrane single ion channel kinetics, was developed. This utilizes a comprehensive microcomputer based system of software with specialized analog and digital electronics for data acquisition, analysis and archiving. Unique discrete signal processing techniques for characterizing the electronic recording system, including specialized hardware and software adapted for minimizing distortions in biosignal recordings, are discussed in detail.

INTRODUCTION

Traditional methods of collecting, analyzing and storing data from cardiac cell membrane electrophysiology experiments have become increasingly cumbersome and unwieldy as experimental protocols have become more sophisticated and complex. In a long protocol requiring hundreds of individual recordings the sheer volume of data becomes overwhelming. Manual analysis of strip chart recordings is labor intensive and subject to operator variability. In addition, ex post facto analysis fails to provide vital information to the investigator while the experiment is in progress. The quantity and variety of laboratory equipment required for most experiments has grown apace with the protocols. At present most laboratories conducting cardiac cell membrane electrophysiology experiments employ traditional analog methods involving storage oscilloscopes and strip chart and magnetic tape recorders, with manually set timers, calibrators and stimulators. A survey of Circulation and Circulation Research for all of 1986 and of Pflugers Archiv for the first half of 1986 revealed that, of 21 papers making use of cardiac action potentials recordings, only three used any form of automated analysis. Of those three, two were from the same laboratory.

This research uses a laboratory microcomputer to control experimental equipment, to record and archive various biosignals such as action potentials, to perform automated analysis of recorded signals

while the experiment is in progress, and to facilitate mathematical modeling of single channel kinetics. A comprehensive software and hardware system for acquiring, analyzing and storing cardiac cell membrane biosignals is described. Results of automated analysis of action potentials are tabulated and examined using a microcomputer data base. Several techniques for identifying the electrical characteristics of the combination of cell/tissue and recording devices are tested and described in detail. The results of this systems identification effort are used to correct the distortions inherent in the biosignals recorded during these experiments. By integrating microcomputers into the cell membrane electrophysiology laboratories and then using them to apply a broad spectrum of computer management, data base and systems science techniques to electrophysiology experiments a considerable improvement over current analog methods is realized.

To achieve these objectives, this research project was divided into three phases:

1. Design and implementation: A hardware/software system was designed to facilitate digital acquisition and analysis of microelectrode, suction electrode and discrete channel recordings. The acquisition, storage and analysis segments were required to be capable of handling very large quantities of data from recordings and analyses. Each single experiment involved as many as one hundred recordings of 100,000 data points per recording. The data base was integrated into the system to store, retrieve and analyze the results of each experiment.

2. Post-processing: Recordings of cell membrane biosignals are inherently distorted due to the electrical characteristics of the cells, the geometry of the electrodes and pervasive use of high-gain electrometers. Post-processing of the recordings improves the quality and signal-to-noise ratio of cardiac electrophysiology recordings by application of digital signal processing techniques. This represents a further extension of the microcomputer into the traditional laboratory setting, and relies on digital methods which are not available in traditional analog compensation schemes.

3. Mathematical modeling: Cell membrane biosignals acquired and processed to improve resolution are used in conjunction with computer-based mathematical simulation to investigate a current problem in cardiac electrophysiology: appropriate modeling of discrete channel activity. Currently available parameters for single channel kinetics derived from previous investigations are used in models of various forms to compare different approaches. This phase takes advantage of the ability of the microcomputer to perform complex and lengthy analyses quickly and efficiently.

CHAPTER I: DATA ACQUISITION, STORAGE AND RETRIEVAL

INTRODUCTION

Although still somewhat rare, microcomputers are becoming increasingly commonplace in electrophysiology laboratories, and several systems have been devised which employ microcomputers in electrophysiology studies (Elharrar and Lovelace, 1979; Fusi et al, 1983; Klein et al 1983; Diez Gil et al 1984). The power and flexibility afforded by the laboratory computer can be used to maximum advantage in an integrated system for acquiring, analyzing, manipulating and storing data from cardiac electrophysiology experiments. Since several types of experiments are conducted in two different physical locations, using a variety of protocols, one general purpose system has been devised and implemented for all types of experiments.

The Data Acquisition and Analysis System (DAAS) is based on the IBM PC/AT and employs commercially available peripheral equipment, with the exception of a specially designed peripheral processing unit (PPU) which coordinates the activities of the laboratory electronics under the control of the PC/AT. The PPU is described in detail in the next chapter. This chapter deals with an overall system of hardware and software used to acquire, analyze and store both transmembrane and voltage clamp data. The log records of experiments and the results of automated analysis of action potentials are stored in a data base

(KnowledgeMan) for statistical analysis of experiments, and for future reference. The digitized data files are kept in archives on removable large capacity disk cartridges. The major components of this system, an IBM PC/AT microcomputer with DT2801A A/D board plus a dual 10-mbyte cartridge disk drive, are off-the-shelf items. For this reason the system is simple and relatively inexpensive to assemble.

HARDWARE

A block diagram of the complete system is shown in Figure 1.1. Centered on an IBM PC/AT, the hardware also includes a dual 10 Mbyte cartridge drive for bulk data storage and an internally mounted A/D board. High reliability and unlimited storage capacity make the Bernoulli Box (Iomega) dual 10 Mbyte disk drives convenient for mass storage of digitized signal recordings and experimental parameters. The data are quickly and easily transferred and archived with dual disk drives.

Voltage clamp or transmembrane action potential waveforms are amplified by an electrometer, then digitized via the Data Translation DT2801A A/D converter which has a maximum sampling rate of 25 kHz. The peripheral processing unit, described in the next chapter, is housed in the same box which contains the terminal board for the DT2801A and both the A/D converter and the peripheral processor are connected to external electronics (electrometer and/or voltage clamp device, sawtooth calibrator and stimulus isolators) by coaxial cables to minimize interference from external sources such as 60 Hz power lines.

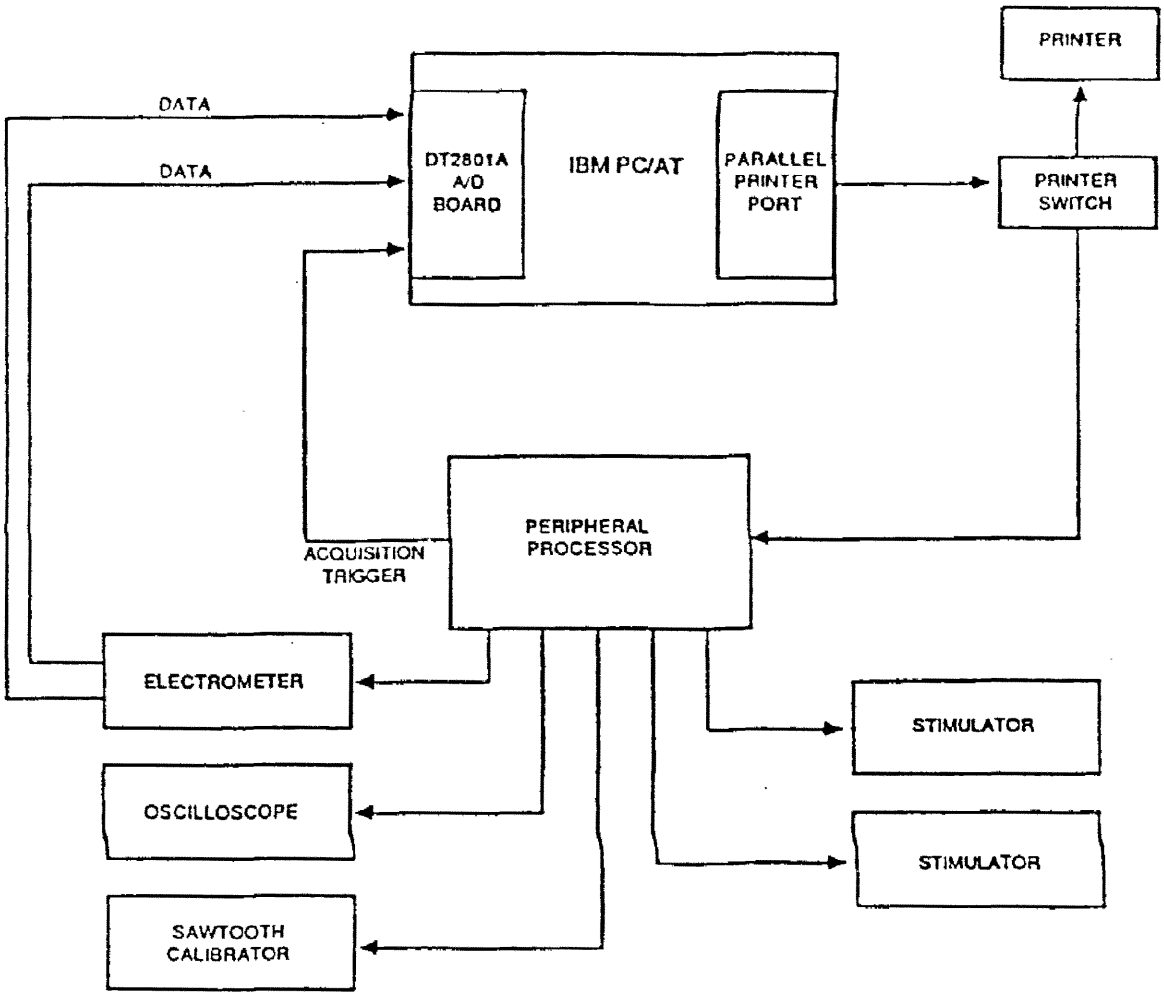


Figure 1.1: The complete hardware layout. The DT2801A A/D board is internal to the IBM PC/AT, while the PPU is housed in a separate box. The remainder of the equipment is mounted in a free-standing equipment rack.

For transmembrane action potential recordings, a World Precision Instruments KS-700 dual channel electrometer permits either single or simultaneous dual channel recordings. Voltage clamp protocols usually involve dual channel recordings of both current and voltage, using either the AXOPATCH-2 or the AXOCLAMP-1A (Axon Instruments). All of the associated electronics are controlled by the peripheral processor which receives commands from the PC/AT via the parallel printer port. The manually-operated printer switch is normally set for the printer and is switched over to the PPU only when a change in experimental protocol is to be effected. This permits downloading of protocol parameters such as

stimulus cycle length to the PPU via the printer port, obviating the need for additional hardware to provide communication between the host computer and the PPU. At all other times, PPU operation is independent of the PC/AT. This hardware configuration allows the experimenter to control many aspects of the experiment from the microcomputer keyboard, and reduces setup time.

SOFTWARE

The core of the system is the software, shown in block diagram form in Figure 1.2. All of the programs used by the DAAS are accessed from one menu and every program returns to that menu.

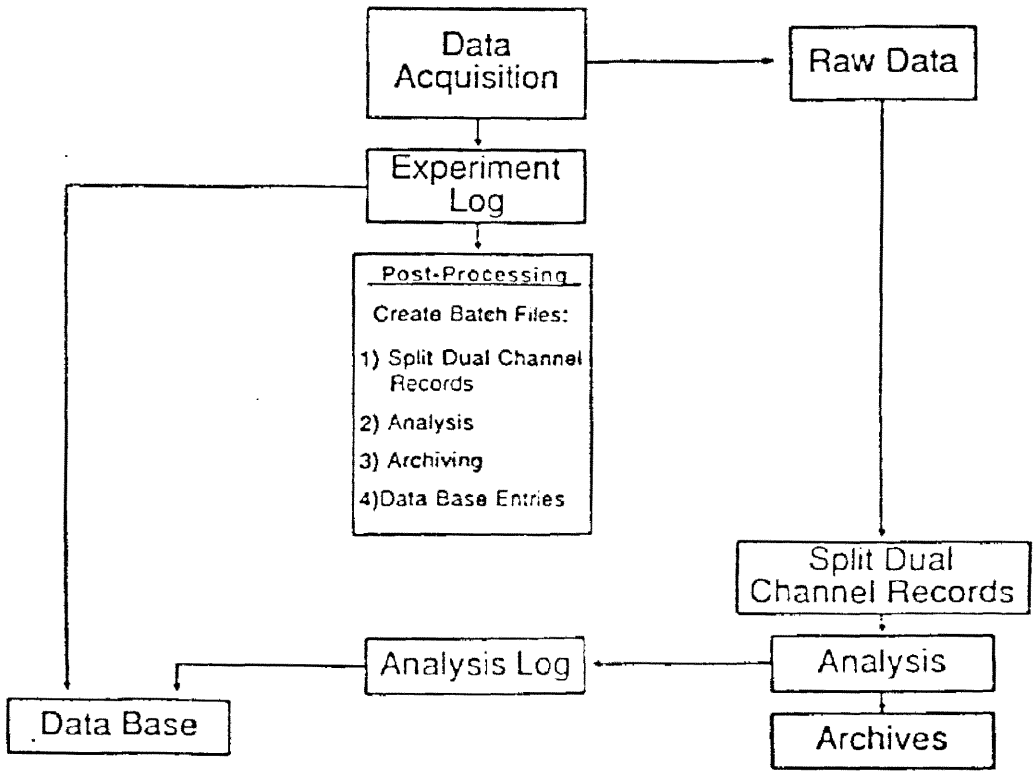


Figure 1.2: Data acquisition, analysis and storage software. All routines are accessed by the operator from a single control menu.

All of the programs are written in Turbo-PASCAL (Borland International) which provides direct access to 640 kbytes of internal memory and compiles and executes very quickly.

ACQUISITION

Data acquisition is carried out by a program titled DATA6D, the flowchart for which is shown in Figure 1.3. This program (1) controls the DT2801A A/D board; (2) sets parameters for the peripheral processor; (3) acquires, displays and stores data and (4) produces a log file of experimental data in a format compatible with the data base. The main screen shown in Figure 1.4 is used to control data acquisition and to change the variables associated with the experiment. When the operator switches from transmembrane action potential mode to voltage clamp mode the "Group A" menu is replaced by the menu shown in the bottom of Figure 1.4. All of the information contained on this main screen is stored at the beginning of each data recording in the file format shown in Figure 1.5. Digitized recordings are accessed on the disk cartridges via PASCAL's BlockRead and BlockWrite procedures, which permit rapid transfer of data in 128 byte blocks. Using the Block procedures, the programs in DAAS write to and read from the disk cartridges at about 200,000 bytes or 100,000 data points per second. File names are generated automatically by the program and consist of a starting character identifying the type of file format, a four-digit Julian date and a three digit serial number. The log file contains the file name, date and time of the recording, protocol parameters (cycle time, stimulus pulse delay and duration, etc.) and information about the experiment such as tissue type, source and age, type and

temperature of solution, treatments which have been applied such as drug perfusions or changes in superfusate ion concentrations and comments.

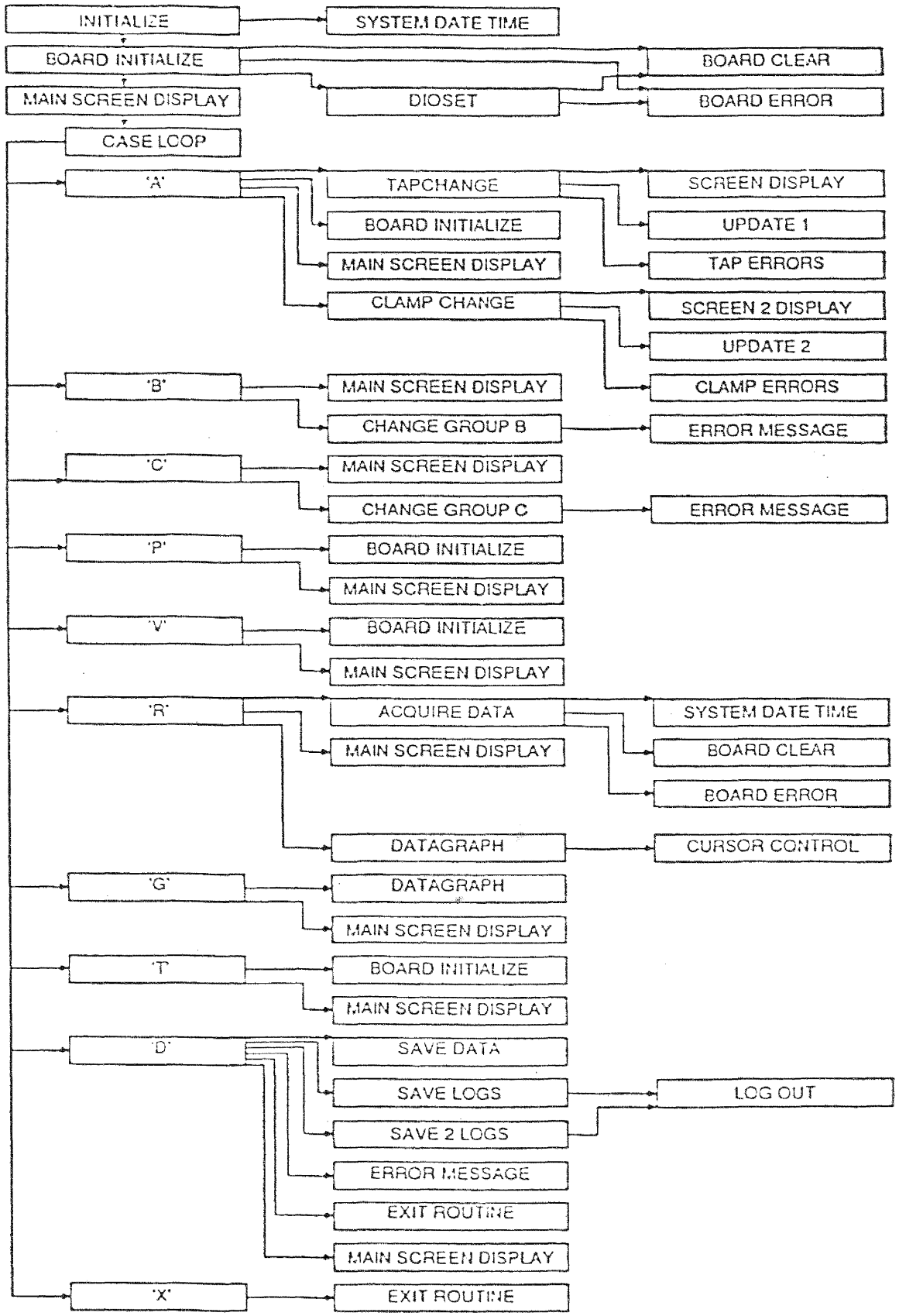


Figure 1.3: Flow chart for the data acquisition program, DATA6D. The functions of the various procedures are described in the text.

POST-PROCESSING

At the conclusion of a set of experiments, the cartridge disk system contains the digitized data files and the log file. The log file is used in the post-processing phase to generate a set of automatically executing procedures or "batch files" which are then used to carry out the operations of analysis, archiving, and attachment of information to the data base. Dual channel recordings are initially stored as interlaced records; the post-processing subsystem splits each dual channel record into two single channel records with ".1" and ".2" file name extensions corresponding to channel one and channel two. After post-processing, the operator may select automated analysis of all of the data files. As each record is processed, a corresponding text file is created which contains the results of the analysis.

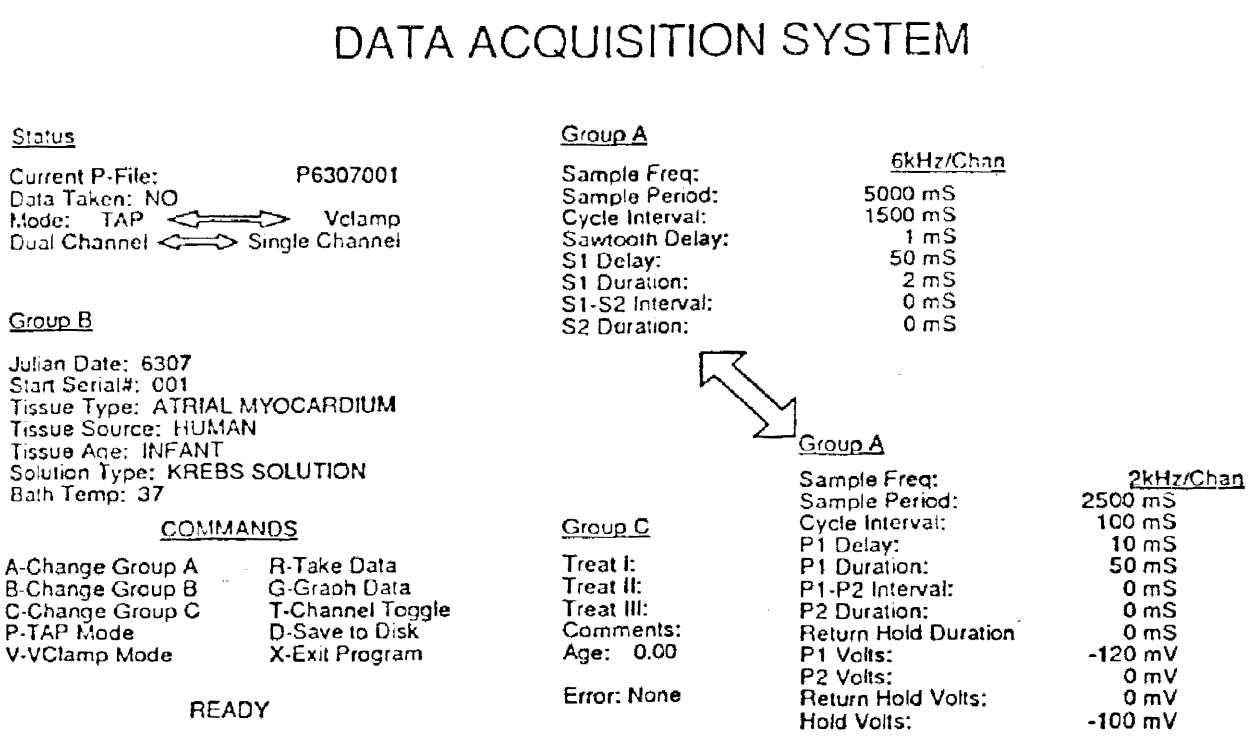


Figure 1.4: Main screen display from the data acquisition program. Information about the experiment is divided into three groups labelled Group A, Group B, and Group C, each of which has its own screen for reviewing and changing the variables.

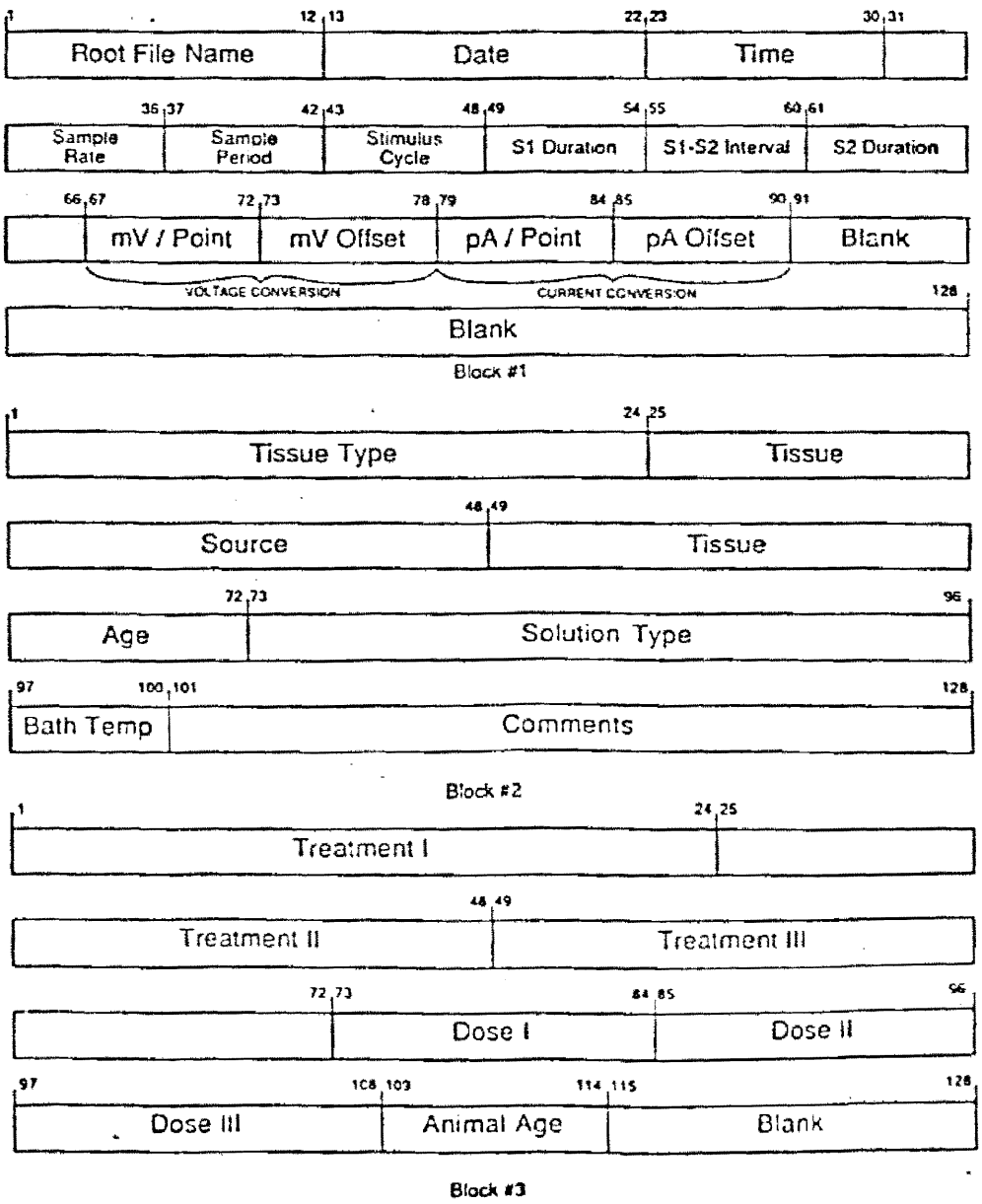


Figure 1.5: Standardized data file format.

AUTOMATED TRANSMEMBRANE ACTION POTENTIAL ANALYSIS

Automated analysis is done using a rule-based algorithm. While doing the measurements by hand, it was observed that all of the commonly referenced parameters could be calculated after three points on each action potential had been located. These points are (1) activation voltage (AV), (2) point of peak overshoot and (3) maximum diastolic potential (MDP). Careful review of many recordings allowed us

to devise a set of "rules" or conditions by which the computer can recognize each of these three points in sequence while scanning a recording in series time sequence. The rules are applied to the digitized recordings to define the three key points as follows:

1) Activation voltage: the slope of the curve as estimated from the recording exceeds a positive threshold. Default value: 0.6 V/s.

2) Peak overshoot: highest magnitude obtained between AV and point where slope of curve becomes negative.

3) Maximum diastolic potential:

a) Slope of the curve in a 60 ms window becomes nonnegative.

b) Occurs at least 50 ms (default value) after peak overshoot. This eliminates false MDP measurements due to the notch between phases 1 and 2.

The default values may be changed interactively by the operator to compensate for unusual waveforms or high noise levels. These rules are implemented based on computing median values from windows of data points which are two milliseconds in duration. Use of the median provides good impulse noise rejection, eliminating errors due to artifacts or noise spikes. The value of 2 milliseconds was chosen based on repeated observations that most noise spikes and artifacts are less than 0.5 ms in duration and therefore would be rejected by the median calculation.

The program based on this algorithm begins by computing the difference in median values between two windows 60 ms apart. If the difference is nonnegative, the beginning of the record is marked as a

valid starting point and a search for AV begins. Otherwise both windows are shifted 2 milliseconds and again compared, looping at this stage until a valid MDP point is found or the end of the record is encountered. Once an MDP is marked, the difference in median values from two adjacent windows is tested against a threshold value to see if an activation point has been found. If not, the windows continue to shift down the record until an AV is encountered. The peak overshoot routine also uses adjacent windows, this time to search for a negative slope. When the peak is found, the program shifts back to the MDP segment and the process repeats until the end of the record is encountered.

Action potential amplitude is computed as the difference between peak overshoot and MDP. This value is then used to locate the APD_{50} and APD_{90} points for each event by scanning the record in reverse time sequence from each MDP point until the median values in 2 millisecond windows corresponding to 50% and 90% repolarization potential, respectively, are found.

$\dot{V}_{max}$ is the maximum rate of change in transmembrane potential attained during the upstroke. In the case of a continuous waveform the point of maximum velocity is defined as the point at which the first derivative reaches a positive peak. Klein et al (1983) approximated the first derivative from a discrete-point recording as the maximum first forward difference between two data points in the segment between AV and peak overshoot:

$$\dot{V}_{max} = f'(t)_1 = (f(t+\Delta t) - f(t)) / \Delta t$$

After employing this method for some time we found that the estimate

provided by the first forward difference was consistently lower than the maximum upstroke velocity indicated by an electronic differentiator, with an error positively proportional to the upstroke velocity. Conte and de Boor (1980) discussed a first differential approximator which uses three data points instead of two:

$$f'(t)_2 = (-3f(t) + 4f(t+\Delta t) - f(t+2\Delta t)) / 2\Delta t$$

Substituting this expression for the first forward difference calculation yielded results that were consistently higher than the electronic standard, and with approximately the same magnitude and proportion of error as the first forward difference method. It seemed logical to try a compromise between the two estimates, and the technique currently in use is the arithmetic average of $f'(t)_1$ and $f'(t)_2$:

$$f'(t)_3 = (-5f(t) + 6f(t+\Delta t) - f(t+2\Delta t)) / 4\Delta t$$

This appears to be entirely satisfactory for upstroke velocities up to at least 800 V/s (Table I).

The time period between the end of one action potential and the beginning of the next is referred to as Phase four. Phase four slope is recorded as the slope between an MDP point and the following AV point. The interval between events is computed as the time between sequential peak overshoot points.

After analysis is completed, a list of parameters for each event is printed, as is a graph of the record with markers for AV, peak overshoot, APD_{50}, APD_{90} and MDP for each event. The measurement data are then put on the disk in a format suitable for attachment to the data base. The data records are transferred to an archives disk, then

erased from the acquisition disk. The log file and analysis results are added to a data base maintained on a separate disk and statistics are compiled from the action potential parameters in the data base.

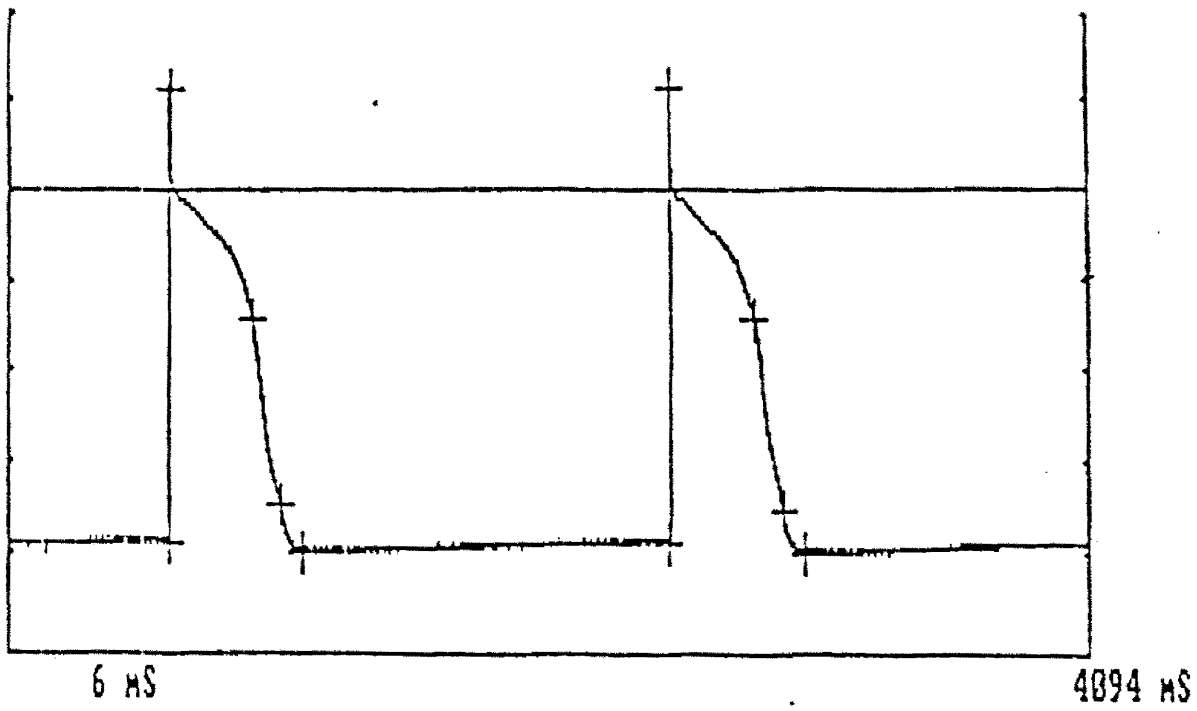


Figure 1.6A: Stimulated action potentials from canine Purkinje fibers, digitized at 13 kHz sampling rate. Crosses are placed by the analysis program for visual verification of correct operation. Vertical tick marks are at 25 mV intervals.

A portion of a typical digital recording of stimulated Purkinje fiber transmembrane action potentials (TAPs) is shown in Figure 1.6a. The TAP parameters are marked by the computer so that the accuracy of the analysis can be judged visually. Reviewing the analysis results by eye is generally sufficient since errors made by the computer tend to be large and obvious. In addition to stimulated recordings, spontaneous Purkinje TAPs are analyzed equally well using the rule-based algorithm,

as are spontaneous TAPs from rabbit sinus node (Figure 1.6b). Action potentials with very slow upstrokes, such as those from the sinus node, may require adjustment of the activation voltage slope threshold parameter (defined in Rule #1) for accurate determination of activation voltage. Also, when dealing with very slow upstrokes, it is often difficult, even for an experienced operator, to decide exactly where the activation points are located. The activation points can be located earlier in time with lower thresholds, but sensitivity to noise increases correspondingly. The default value of 0.6 V/s is a compromise which generally provides accurate location of activation point and good noise immunity.

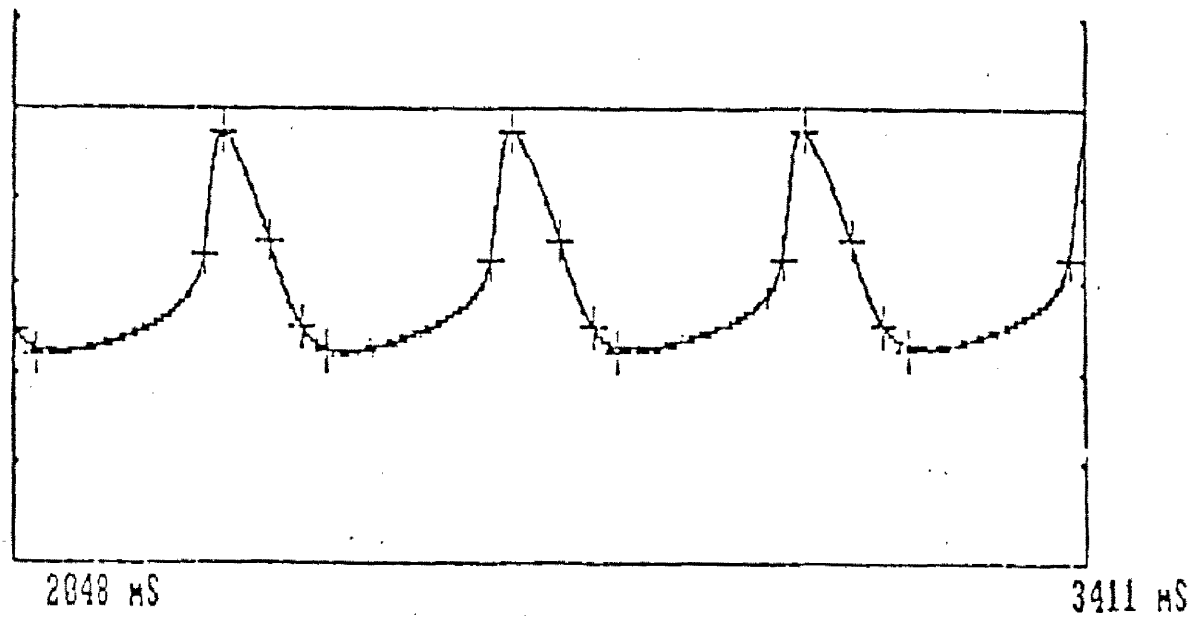


Figure 1.6B: Spontaneous action potentials from rabbit sinus node, digitized at 2 kHz. Vertical marks are at 25 mV intervals.

VALIDATION OF AUTOMATED TAP ANALYSIS

Measurement of action potential parameters is commonly done by hand from strip chart recordings or from photographs of the waveform displayed on a storage oscilloscope. As this is a widespread practice, hand measurement is a reasonable standard against which to validate the automated analysis algorithm.

Canine Purkinje fibers were superfused with standard bicarbonate buffered Krebs solution (containing 4.0 mM K^+ and 2.7 mM Ca^{++}) saturated with 95% O_2-5% CO_2, and warmed to 37 C. The tissue was impaled with glass capillary microelectrodes filled with 3.0 M KCl, and having tip resistances of 10-20 megohms. The transmembrane potential was amplified by a capacitance neutralization electrometer. TAPs were recorded digitally (13 kHz sampling rate) by the data acquisition system described in the next chapter and simultaneously displayed on a storage oscilloscope (Tektronix 5111A). The TAP upstrokes were electronically differentiated and displayed on the B time base of the oscilloscope at a rapid sweep speed (0.5 ms/div) using an analog differentiator calibrated with sawtooth pulses of known ramp velocity (Bigger et al 1965). To avoid error due to nonlinearity of the differentiator, the calibration signal slope was always adjusted to within 100 V/s of the action potential upstroke velocity. For each recording, 35mm photographs of the oscilloscope screen were taken and then printed on 8x10 inch paper to maximize the accuracy of the hand measurements. Transmembrane action potential measurements were made with the aid of a digitizer pad (Houston Electronics) with a resolution of 0.1 mm. Ten randomly selected TAPs recorded using both analog and

digital methods were analyzed, with results listed in Table I. These results indicate excellent agreement between the two methods. Paired t-tests at the 5% level of significance failed to reject the hypothesis of no difference between the data sets for all parameters. In the case of the data for $\dot{V}_{max}$, however, it is noted that the variance of the differences between the values estimated by the electronic differentiator and by difference calculations is fairly large, possibly reducing the power of the paired t-test significantly. To verify the strength of the relationship between the values calculated by the two methods the coefficient of determination, r^2, was calculated from the data pairs. The computed value of $r^2 = 0.9961$ indicates that, based on these ten data pairs, the difference calculation method is an excellent predictor of the $\dot{V}_{max}$ values estimated by the accepted standard, electronic differentiation.

DATA BASE

This subsystem employs the Knowledgeman data base manager (Micro Data Base Systems, Inc.) The ASCII (text) log file created by the acquisition program contains information about each data record, in a file format which is compatible with the data base file reading system. Results of the automated analysis routine are also in compatible files. Post-processing of a set of experiments produces a Knowledgeman procedure which attaches the log and analysis files to the appropriate tables in the data base. After verifying proper data transfer, all data, log and analysis files are erased from the acquisition program disk.

Formats for the data tables are shown in Figure 1.7. TAPS contains information about the experiments, such as date and time the record was taken, digitization rate, sample period, tissue identification information and treatments and doses applied to the tissue or cells. Results from automatic analysis of action potentials (amplitude, MDP, $\dot{V}_{max}$, etc.) are stored in TAPTAN. A unified table, TAPUT, which combines information from both sources is created using the temporary files TAPX and TAPTANX. TAPUT provides simple and rapid cross-correlation of data for analysis.

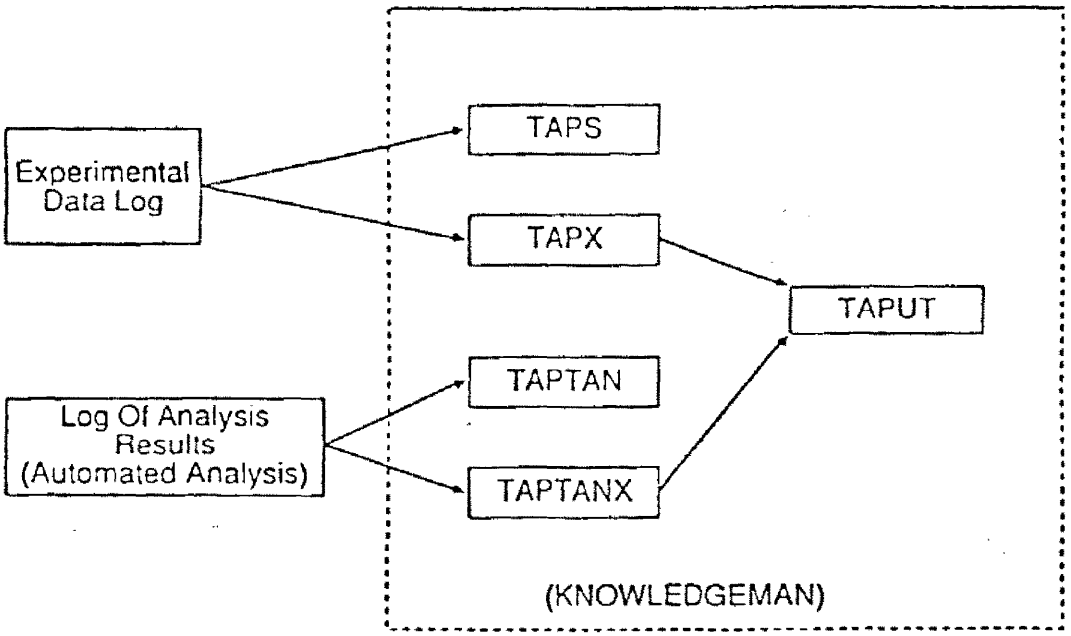


Figure 1.7: Layout of the data base tables. TAPS contains information pertaining to the experimental setup and protocols, while TAPTAN holds the results of automated analysis of action potential recordings. TAPX and TAPTANX are temporary tables which are used to create a unified table, TAPUT, and are then discarded.

DISCUSSION

The DAAS approach to digitizing and storing cardiac TAP data differs somewhat from previous computer-based methods. Due to memory constraints in earlier microcomputers a dual frequency sampling protocol was frequently employed in which the highest frequency components in the TAP data (in the upstroke, between AV and peak overshoot) are sampled at the maximum rate available and lower frequency components in the remainder of the TAP digitized at a much slower rate (Elharrar and Lovelace, 1979; Klein et al, 1983). In these systems the TAP data are immediately analyzed and in general only the TAP parameters are stored in tabular form because of the relatively limited permanent storage capacity of floppy disks. Our approach is to record, digitally, one or two channels of TAP analog data at a continuous sample frequency which is sufficiently fast to capture these data with fidelity (Buckles, Hewett, 1986). The sample period is adjusted to capture 3-5 TAPS in each recording and the basic cycle length of the stimulated TAPS or the automatic rate in spontaneous preparations determines the data sample period. This method is more easily implemented than in the past because the present generation of microcomputers has adequate addressable random access memory (640 kbytes or greater) and mass storage devices on which these large records can be stored.

Another factor which makes a constant digitization rate practical is that, based on experiments conducted in these laboratories, the minimum sample frequency required to record a Purkinje fiber V_{max} in

the range of 700 to 800 V/s with fidelity is a surprisingly low 13 kHz (Buckles, Hewett, 1986). Mathematical estimation has proven to be a satisfactory method for determining the highest $\dot{V}_{max}$ values thus far recorded, approaching 800 V/s. This was determined by comparing measurements made with a carefully calibrated electronic differentiator and those calculated by computer.

Several advantages are derived from digitization of TAP recordings at a constant rate and ex post facto analysis: 1) The recording process is somewhat simpler because the experimenter need not be concerned that the rapid sampling window coincides with the TAP upstroke. 2) The digital TAP recordings are easily copied and the copies can then be filtered or edited as needed. Part of the record can be blanked or truncated. Retaining the original record in archives affords security against lost data because analyses can be repeated and TAP parameter data regenerated if necessary. 3) The TAP analysis can be done after post-processing and can be repeated as many times as needed to obtain satisfactory results.

The disadvantage of this approach is that the results of TAP analysis are not immediately available for evaluation and to provide feedback during the experiment. As currently implemented, TAP analysis of a 121 kbyte record requires about 55 seconds. This is not an excessive amount of time and for many experiments where the intervals between data recordings are sufficiently long, the acquisition program is exited and TAP data analysis is run while the experimenter is waiting to make more recordings. Aside from the processing time restriction, the main hardware limitation of this system is the time

required to display large data files on the screen. Graphic display on the CRT of every point in a very long recording can take several minutes. This limitation has been overcome to some extent by interfacing the microcomputer with a Gould strip chart recorder which provides hard copy of a recording in a few seconds.

Integration of the data acquisition system and a data base manager permits easy and rapid retrieval and manipulation of the stored TAP parameters for tabulation, statistical analysis and display. For example, the results of the experiments can be sorted to produce statistics showing effects and interactions. This also permits establishment of a set of norms with respect to inter- and intra-recording beat-to-beat variability in the data. This provides a statistical frame of reference for determining when the variability in the data is due to the actual presence of effects or can be attributed to random or experimental error. It has been noted that within a single record containing multiple stimulated action potentials from normal Purkinje fibers, variations in APD_{50} and APD_{90} average less than one percent, while upstroke velocity shows much greater variability. Treatments with drugs or ions may produce much more beat-to-beat variability in these parameters, requiring a larger number of action potentials for statistical analysis.

The convenience and efficiency of this completely electronic approach to data storage and processing is dependent to some extent on the experimenter's proficiency in using the data base command structure. The utility of the data base has been enhanced with a number of standardized procedures and utility programs within the data base

which perform many frequently required tasks. The usefulness of the system is particularly evident when a large amount of complex data is collected, such as with a long TAP pacing protocol with several different treatments. In experiments such as whole cell voltage clamps, often the amount and diversity of data is much less than in TAP experiments, lessening the contribution of the data base.

CHAPTER II: PERIPHERAL ELECTRONICS FOR DATA ACQUISITION

INTRODUCTION

As discussed in the previous chapter, experiments in cardiac electrophysiology have benefited from increasing availability and sophistication in laboratory microcomputers. Complete systems centered around a microcomputer have been designed and successfully implemented in several laboratories (Elharrar and Lovelace, 1979; Fusi et al 1984; Diez Gil et al 1984; Klein, Jenkins and Ten Eick 1983). The computer system is usually interfaced to the environment by an analog-to-digital (A/D) board, of which several are commercially available. Peripheral electronics provide synchronizing pulses for stimulus generation, oscilloscope sweep, data display and recording and calibration signals. In many cases, however, addition of a computer to an already complex array of electronic equipment increases the level of difficulty encountered in setting up and running an experiment. In complex experimental protocols smooth and efficient coordination of all of the required equipment using TTL timers and clocks is often difficult. Interconnections between equipment become dauntingly complicated, increasing the likelihood of error or equipment failure. Hence the experimenter is required to set or dial in all parameters by hand for each change in protocol. In addition, the likelihood that all equipment and connections will function together at a time chosen by the

experimenter decreases with increasing complexity.

These difficulties can be circumvented by a device designed specifically to control laboratory electronic equipment and which is programmed by the laboratory microcomputer. This chapter describes a dedicated microprocessor-based peripheral processor (PPU) which produces timing and control pulses of appropriate magnitude and duration as instructed by, but independent of, the host microcomputer. This arrangement permits the experimenter to control all of the various peripheral instruments from the laboratory computer keyboard without significantly affecting the data acquisition or analysis capabilities of the host computer.

HARDWARE

The arrangement of the system, discussed in the previous chapter and shown in Figure 1.1, highlights the central role played by the PPU in coordinating the activities of the various electronics. Parameters for the experimental protocol are entered into the laboratory computer, an IBM PC/AT, using a PASCAL routine which formats the variables for the PPU. A printer switch allows both the printer and the PPU to be connected to a single parallel printer port on the microcomputer. Communication between the laboratory computer and the PPU takes place only at initialization, or when a change is effected.

Peripheral processor logic is contained in four large-scale integration chips, interconnected as shown in Figure 2.1. Central to the device is the Motorola MC6802 microprocessor. This chip is a basic

6800-series unit with an on-board 128-byte static random access memory (RAM) memory added. Variables which are accessed during execution of the routines stored in the erasable programmable read only memory (EPROM) are held in the RAM, which is addressed as the lowest 128 bytes in memory. The stack pointer and interrupt vectors are also kept in the RAM.

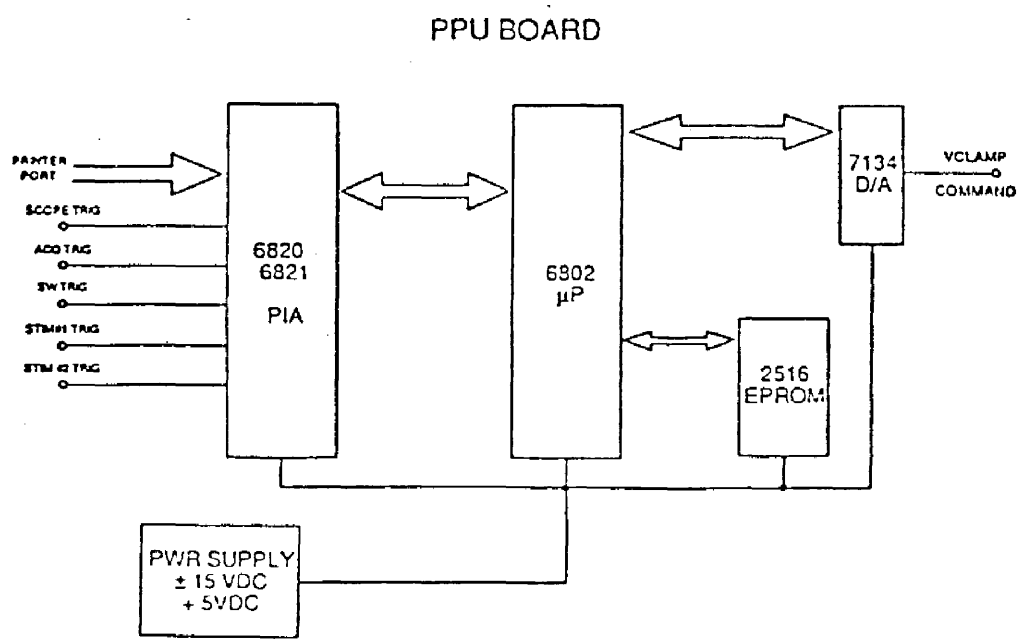


Figure 2.1: Schematic of the PPU board, showing main interconnections between the major integrated circuits. The +/- 15 VDC lines provide power for operational amplifiers used as current drivers.

The TMS2516JL-45 EPROM contains several machine language subroutines for experimental protocols and is hardwired on the board so that the first address in the EPROM corresponds to microprocessor address 8000 (hexadecimal). The current version of the PPU EPROM software is listed in the Appendix.

The Intel 7134A D/A chip is a 14-bit current converter with two operational amplifiers and support circuitry arranged to provide command voltages for voltage and current clamp experiments. Command voltages for the D/A converter are divided into high and low 8-bit words and stored in the MC6802 static RAM. The Axopatch-1A used for clamp experiments has an external clamp command sensitivity of 20mV per volt, and the +/- 15 volts available from the 7134A provide a clamp range of +/- 300 mV.

A Motorola MC6820/6821 Peripheral Interface Adaptor (PIA) has two eight-bit input/output ports. One port links the PPU with the PC/AT via an 8-bit port, and the second port is used for output of timing and trigger pulses. Output lines to the sawtooth calibrator and the stimulus isolators are conditioned by a set of 741-class operational amplifiers wired as low impedance drivers to match the requirements of the equipment.

The four main chips and associated electronics are mounted on a 4.5 x 4.5 inch circuit board with a 44 pin card edge connector, and plugged into a socket mounted in an aluminum box. Pin assignments on the card edge connector are listed in Table II; the pins are numbered from left to right looking at the component side of the board with the connector at the top. The mounting box also contains power supplies which provide +/-15 VDC and +5 VDC. A list of parts is given in Table III. Point-to-point wiring for the PPU, which is rather complicated due mainly to the data and address busses, is displayed in its entirety in Figure 2.2.

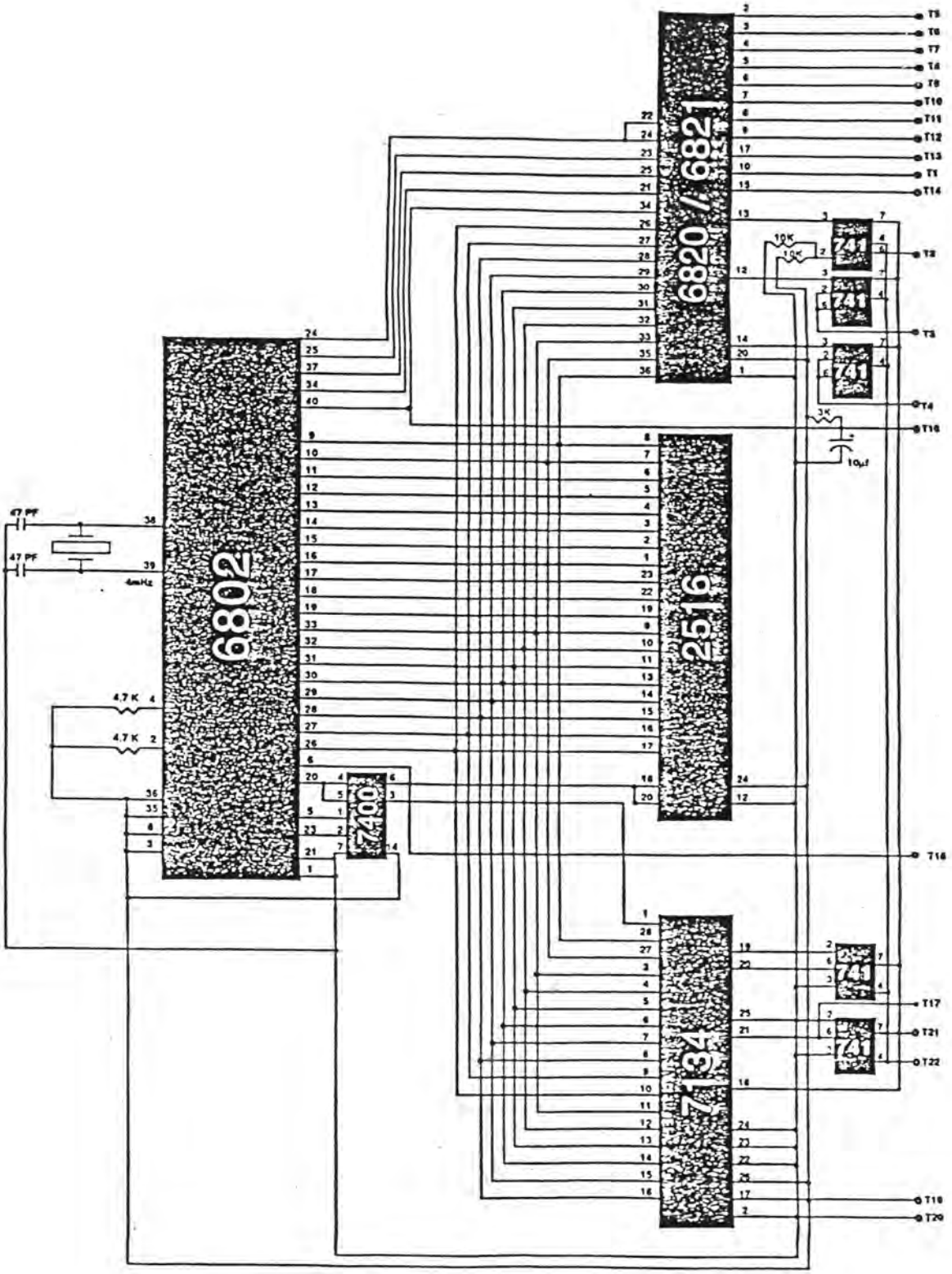


Figure 2.2: Point-to-point wiring diagram of the PPU board, showing all interconnections and support components. The power supplies are external to the PPU board and are not shown. Tab numbers on the right side refer to pin assignments listed in Table II.

SOFTWARE

PPU software is in two contiguous segments: a control and communications routine for the IBM PC/AT and the machine language program contained in the EPROM. The requirements for formatting and sending variables to match the PPU EPROM software are listed below and any program on any host computer which meets these requirements will suffice. Our control computer program is written in Turbo PASCAL and provides a convenient format for entering or changing a protocol.

The EPROM program initializes the PPU on powerup, accepts variables from the host computer and provides two major classes of experiment protocols. Transmembrane action potential recordings are handled with the routine shown by flowchart in Figure 2.3 and a similar flowchart for the voltage clamp control program is given in Figure 2.4. Upon startup or reset an initialization process defines a set of default parameters and performs routine organizational chores such as setting the stack pointer and interrupt handling vectors. Variables are downloaded to the PPU by an interrupt service subroutine when a non-maskable interrupt is received from the host computer, a process which requires a total of approximately 500 ms to complete. At all other times the PPU operates independently of the host computer and therefore can be used for tissue conditioning or maintenance while the host computer is occupied with other tasks such as analyzing data previously obtained.

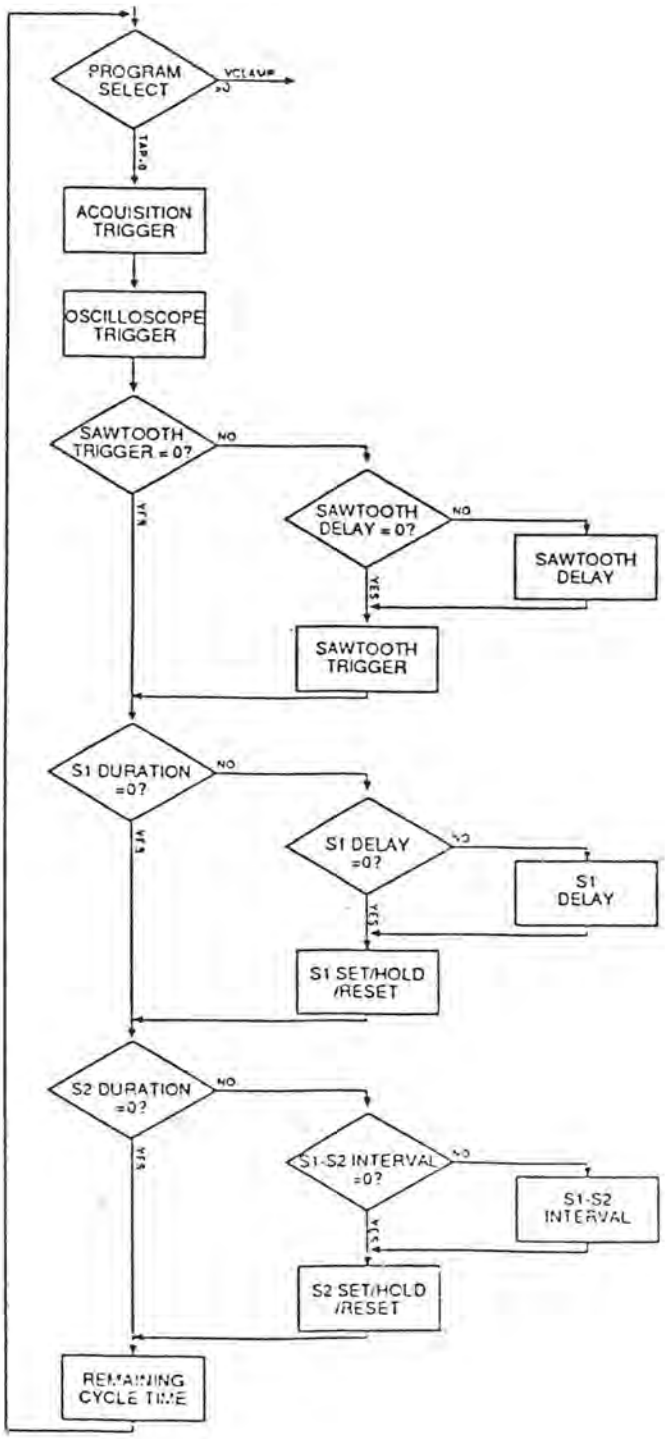


Figure 2.3: Flow chart of the transmembrane segment of the PPU EPROM program. A complete listing of the assembly and machine language software is provided in the Appendix.

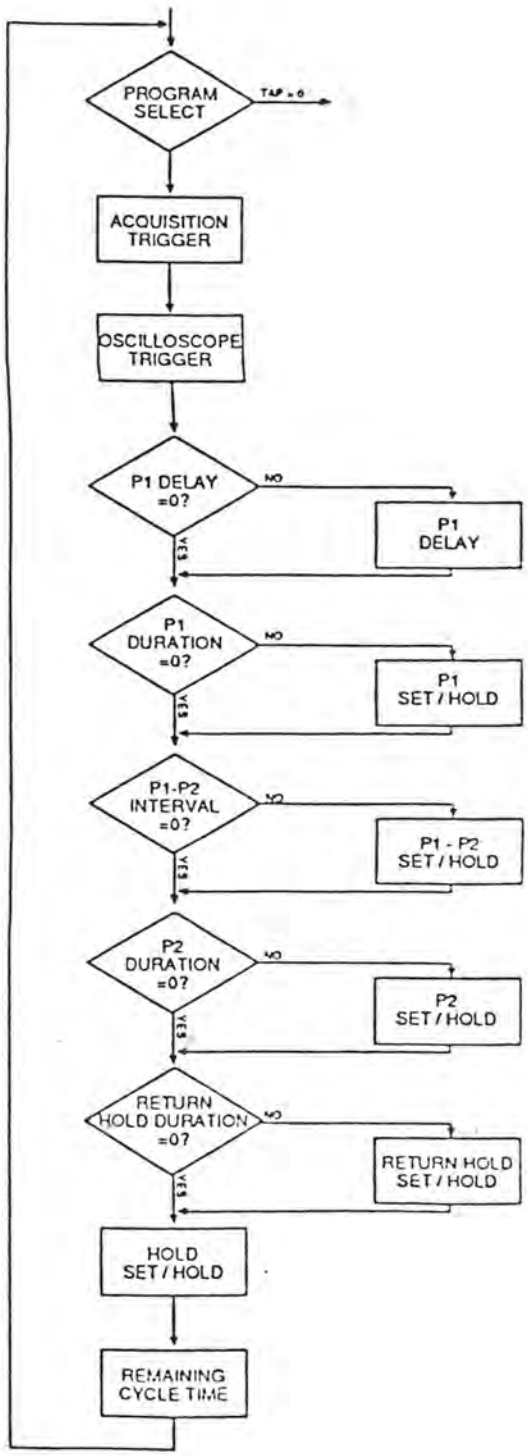


Figure 2.4: Flow chart of the voltage clamp segment of the PPU EPROM program.

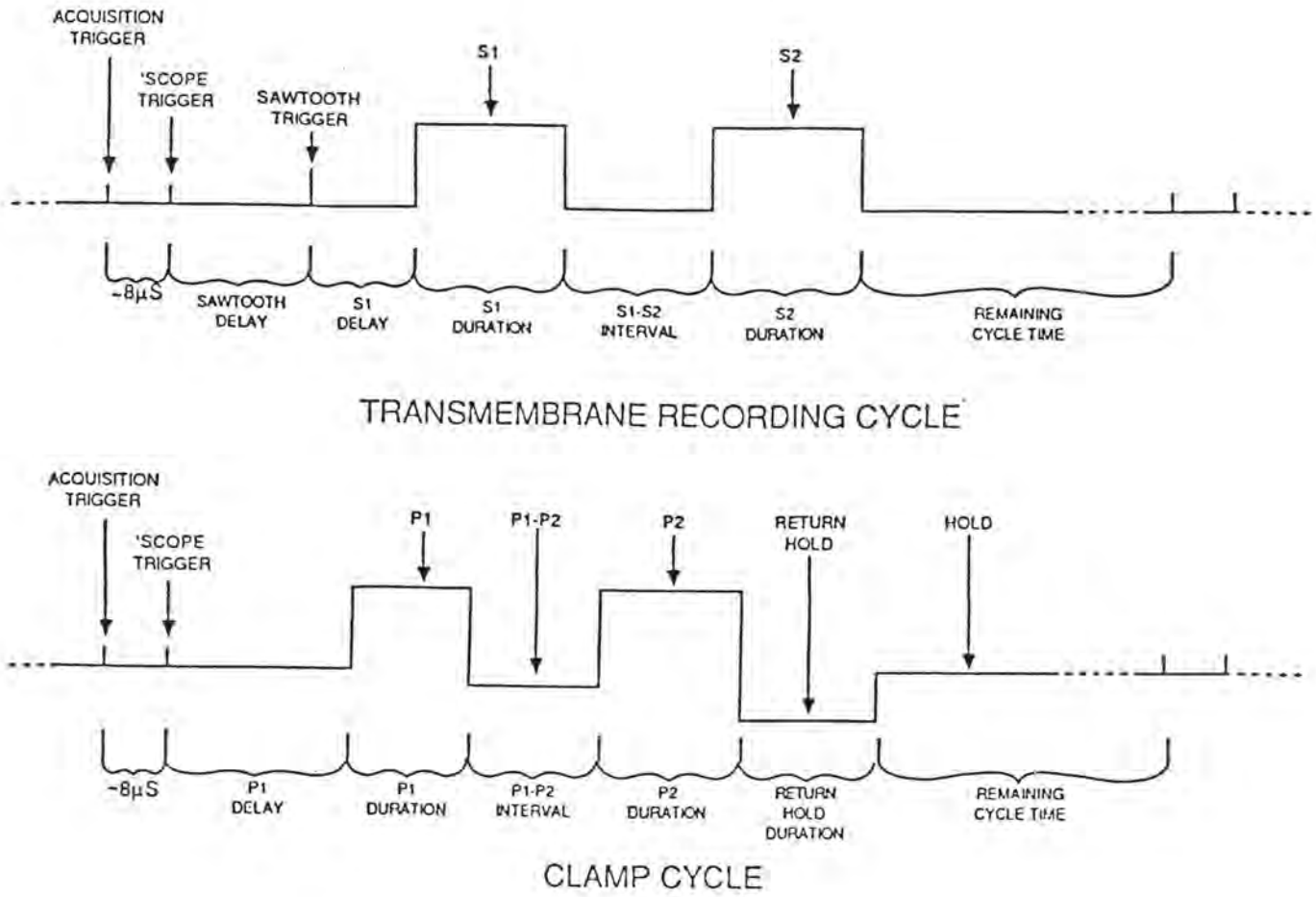


Figure 2.5: Timing cycles for each mode of operation, showing temporal relationships between the various triggers, delays and intervals. Each pulse and trigger can be selected or deselected by the host computer controlling software.

The present generation of PPU software, version 5.01, provides the transmembrane and clamp experiment protocols depicted in Figure 2.5. Note that the stimulus pulses S1 and S2 in the transmembrane section are software selectable to either or both of the dual channel stimulus isolators. Although fairly simple, these routines are very flexible and can easily be modified for more complex protocols either by changing the EPROM programming itself or by writing the host computer software to change the parameters automatically at different stages of an

experiment. In the transmembrane action potential segment, for instance, it is occasionally desirable to run an experiment with an S1-S2 interval which decrements or increments to a desired maximum or minimum. This can be accomplished without modifying the present EPROM software by periodically downloading appropriate changes in variables from the host computer.

HOST COMPUTER INTERFACE

Twenty-seven variables are downloaded from the host to the PPU. Each is one 8-bit word, sent in parallel to one of the ports on the Motorola MC6820/6821 PIA. Data transfer is initiated when the host computer brings the nonmaskable interrupt line low for 10 milliseconds, causing the MC6802 to enter the interrupt server subroutine. Eight data bits are placed on the connecting bus and then strobed into the PPU via the data transfer strobe bit. The variable names, functions and definitions are listed below and all times are given in milliseconds.

1. PRGSEL — Program selector: 0 = Transmembrane routine, 1 = Clamp protocol.
2. SCTRIG — Oscilloscope trigger code. One of the 8-bit ports on the PIA is used as a set of output lines. The code determines which of the lines is set or reset. For the oscilloscope, the trigger is wired to bit 0 on the PIA port so that when a hexadecimal '01' is sent to the PIA the oscilloscope sweep is triggered. Using variables as

trigger codes allows the operator to select or deselect the trigger and permits redefinition by software of the functions of the output lines. The same principle applies to the other output lines.

3. SWTRIG — Sawtooth calibrator trigger. Hexadecimal code is '08'.

4. S1TRIG — S1 stimulus pulse trigger. Hexadecimal code is '04'.

5. S2TRIG — S2 stimulus pulse trigger. Hexadecimal code is '10'.

6. SWDEL — Time interval between SCTRIG and SWTRIG. Maximum is 255 ms.

7. S1DEL — Time interval between SWTRIG and beginning of S1, or between SCTRIG and beginning of S1 if SWTRIG is set to zero. Maximum is 255 ms. Labelled P1DEL in clamp mode.

8. S1DURH — Upper eight bits of S1 stimulus. Calculated as

S1DURH = INT((S1Duration) / 256)

Labelled P1DURH in clamp mode. Maximum duration is 65535 ms.

9. S1DURL — Lower eight bits of S1 stimulus. Calculated as

S1DURL = (S1Duration) - (S1DURH * 256)

Labelled P1DURL in clamp mode.

10. S1S2IH — Upper eight bits of S1-S2 interval. Calculated as

S1S2IH = INT((S1S2Interval) / 256)

Labelled P1P2IH in clamp mode. Maximum interval is 65535 ms.

11. S1S2IL — Lower eight bits of S1-S2 interval. Calculated as

S1S2IL = (S1S2Interval) - (S1S2IH * 256)

Labelled P1P2IL in clamp mode.

12. S2DURH Upper eight bits of S2 stimulus. Calculated as

S2DURH = INT((S2Duration) / 256)

Labelled P2DURH in clamp mode. Maximum duration is 65535 ms.

13. S2DURL Lower eight bits of S2 stimulus. Calculated as

S2DURL = (S2Duration) - (S2DURH * 256)

Labelled P2DURL in clamp mode.

14. RHDURH Upper eight bits of Return Hold duration. Calculated as

RHDURH = INT((Return Hold Duration) / 256)

Maximum return hold duration is 65535 ms.

15. RHDURL Lower eight bits of Return Hold duration. Calculated as

RHDURL = (Return Hold Duration) - (RHDURH * 256)

16. RCYCH Upper eight bits of remaining cycle time. Total remaining cycle time in transmembrane mode is calculated as

RCYCLE = (Cycle time)-(SWDEL+S1DEL+S1S2INTERVAL+S2DUR)

or, when the S1-S2 interval is zero, as

RCYCLE = (Cycle time)-(SWDEL+S1DEL+S1DUR+S2DUR)

In the clamp protocol, remaining cycle time is calculated as

RCYCLE = (Cycle time)-(P1DEL+P1P2INTERVAL+P2DUR +RHDuration)

or, when the P1-P2 interval is zero, as

RCYCLE = (Cycle time)-(P1DEL+P1DUR+P2DUR+RHDuration)

17. RCYCL Lower eight bits of remaining cycle time. Calculated as

RCYCL = (RCYCLE) - (RCYCH * 256)

The remaining variables are 16-bit representations of clamping potentials. With +/- 15 VDC power supplies, the clamping command potential range is approximately +/- 250 mV, calibrated for the AXOPATCH-1A command requirement for 20 mV output for 1 V command input. The general formula for calculating the integer from which the 16 bits are derived is as follows:

Integer = ((15110 - (ClampVolts * 27.6)) * 0.98) + 436

for positive clamping potentials, and

Integer = ClampVolts * 27.6

for negative potentials. This calculation is required in part because the Intel 7134A D/A chip is a 14-bit converter, loaded with a 6-bit word and then an 8-bit data word. The constants in the formula have been empirically determined for our particular combination of power supplies and 741-class op amps so that the clamp potential commands are accurate. Our experience is that each combination of power supplies, operational amplifiers and 7134 D/A chips requires unique calibration constants for optimal precision and accuracy.

In all of the following variables the calculated Integer is split into upper and lower 8-bit words by the same formula used for the upper and lower cycle time and interval words.

18. P1VH P1 potential upper 8 bits.

19. P1VL P1 potential lower 8 bits.

20. P12VH P1-P2 interval potential upper 8 bits.

21. P12VL P1-P2 interval potential lower 8 bits.

22. P2VH P2 potential upper 8 bits.

23. P2VL P2 potential lower 8 bits.

24. RHVH Return hold potential upper 8 bits.

25. RHVL Return hold potential lower 8 bits.

26. HVH Remaining cycle holding potential upper 8 bits.

27. HVL Remaining cycle holding potential lower 8 bits.

DISCUSSION

The primary purposes of counter/timers in most transmembrane action potential experiments are to produce the S1 (or S2) stimulus pulses and trigger an oscilloscope and sawtooth calibrator. This is necessary so that the action potential appears in an appropriate position on the oscilloscope and is immediately preceded by the sawtooth signal. The sawtooth signal is used to calibrate the electronic differentiator (Bigger et al 1968) and must be precisely timed so that the differentiation of the sawtooth and the upstroke can be seen together at a rapid sweep speed on the B time base. In this system the electronic differentiation is not recorded; V_{max} is estimated by a finite-difference calculation from the digitized transmembrane action potential recording as discussed in the previous chapter. Because the minimum acceptable digital sampling frequency is proportional to $\dot{V}_{max}$ (Buckles, Hewett 1986), the electronic differentiation of the upstroke, which provides immediate indication of $\dot{V}_{max}$, is monitored to ensure that the sampling rate is fast enough to capture the upstroke with good fidelity.

The counter/timers on some commercially available pulse generators are able to perform most of the functions incorporated into the PPU.

One A/D board (TECMAR, LABMASTER) has three integral timers which can be programmed to provide synchronization pulses. Brown and Millecchia (1983) designed and built a laboratory controller for neurophysiology experiments which is capable of providing elaborate timing sequences with good temporal resolution, but requires reprogramming of the on-board EPROMs to change the timing sequences.

There are several advantages inherent in the PPU which are not likely to be found in a simple pulse generator or timer:

1) Downloading experimental parameters from the host computer permits the experimenter to control the experiment from the same keyboard which controls data acquisition and analysis. Once the experiment begins, little or no turning of dials and flipping of switches is necessary.

2) The experimental parameters can be stored as files or sets of files in the host computer and reloaded into the PPU as desired. This is particularly useful for designing, retaining and reproducing complex pacing protocols.

3) The device is very flexible. With appropriate host computer and EPROM software modifications, virtually any pulse sequence likely to be required can be generated.

4) A single keystroke changes the mode of operation from transmembrane action potential to voltage clamp and from voltage clamp to transmembrane mode.

The PPU is designed to control voltage clamp experiments through the external voltage clamp command input of the voltage clamp amplifier. Voltage clamp command output from the PPU has proven to be

highly accurate and very stable. The absolute magnitude of the command voltage is dependent on the individual characteristics of the power supplies and the operational amplifiers and calibration of the host computer software is required for each processor and associated power supply. This is done empirically via the host computer software for each peripheral processor, but once completed, recalibration has not been necessary. Our calibrations, which we performed once and occasionally check, have resulted in clamp command accuracy of +/- 0.5 mV of the desired potential throughout the allowable voltage range.

The PPU has several limitations inherent in the design. Signal and trigger lines are conditioned with operational amplifiers to match our specific laboratory setup. Some hardware changes to the PPU may be required if peripheral electronics different than those discussed in this text are employed, although the changes should not be difficult to implement. In point of fact one of these PPUs is in operation at the Department of Physiology, College of Physicians and Surgeons, Columbia University, New York, N.Y. and has been slightly modified to suit the peripheral equipment at that location.

As presently programmed, all times are in increments of 1 millisecond. This permits delays of as much as 65 seconds between events in a cycle, but prevents the use of fractions of milliseconds in specifying the timing of cycle events. In the case of time delay intervals such as that between the oscilloscope trigger and the onset of the first stimulus pulse, this can be inconvenient at high oscilloscope sweep speeds. All things considered these limitations do not seriously affect the utility or flexibility of the PPU.

In summary, the PPU is a microprocessor-based pulse and voltage generator which controls and coordinates voltage clamp external command inputs, stimulators, and monitoring and calibration devices. This unit is programmed by and otherwise operates independently of the host computer. Virtually any computer with a parallel communications or printer port can be used as the host computer. A complete listing of the EPROM assembly language software is provided in the Appendix. Although not required for proper operation, the PPU functions well as part of the digital data acquisition, analysis, storage and data base system described in the previous chapter.

CHAPTER III: DIGITAL TECHNIQUES FOR IMPROVEMENT OF DATA RESOLUTION

INTRODUCTION

Voltage clamp experiments, particularly the single channel patch clamp techniques developed by Neher and Sakmann (1976), provide direct observation of membrane ion channels from many diverse preparations ranging from squid giant axons to isolated ventricular myocytes. The principle of whole-cell voltage clamping is simple: A single electrode is attached to or inserted through a cell membrane and sufficient current of appropriate polarity is passed through the electrode to maintain a stable transmembrane potential. The current required to maintain transmembrane potential is referred to as "feedback current" and is equal in magnitude to the sum of the currents flowing across the membrane from all sources. In the classic Hodgkin and Huxley experiment series (1952) the feedback current was determined to be equal to the algebraic sum of the potassium and sodium currents flowing across the squid axolemma. Cardiac cells typically present a much more complex system in that calcium currents also contribute to changes in membrane potential and there may be several types of channels for each ion species. Also, the geometry of heart cells is less advantageous for good spatial voltage homogeneity. Single channel recording techniques are being exploited to separate and classify the apparent profusion of channel types.

The circuit diagram for a typical whole-cell clamp experiment is shown in Figure 3.1a. The voltage clamping function is provided by the operational amplifier through the feedback resistor. V_{ref} is the desired transmembrane potential. Since the op amp also functions as a current-voltage converter in that (V_B - V_{ref}) is proportional to the feedback current, monitoring that voltage difference is equivalent to monitoring the flow of current across the membrane.

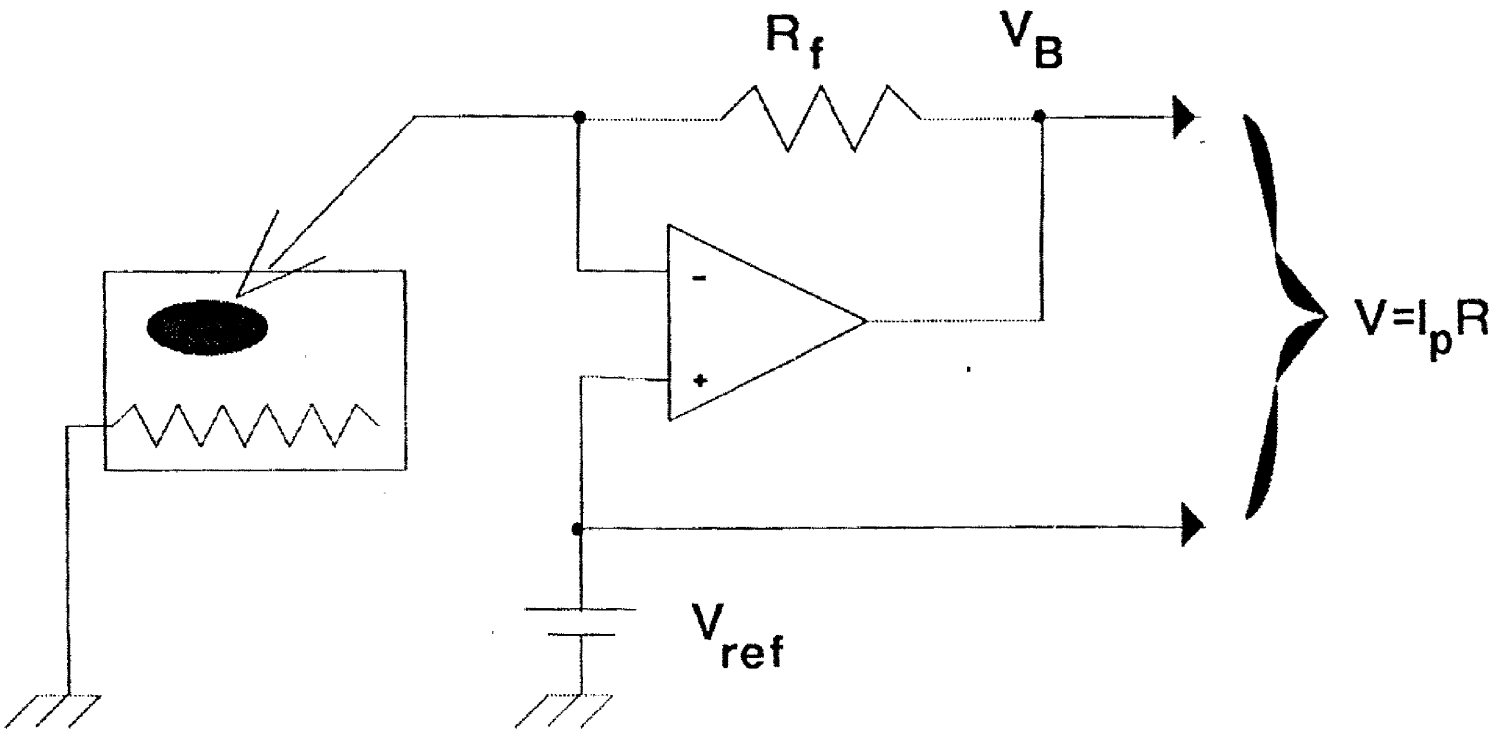


Figure 3.1A: Schematic representation of a voltage clamp experiment. The cell is shown with an electrode attached and a reference electrode is in the bath below the cell. Also shown is the clamping operational amplifier with feedback resistor and reference (clamping) voltage source.

The combination of the cell, the electrode and the clamping circuit can be modeled as an electrical circuit with lumped parameters, as in Figure 3.1b. With some changes in parameters the form of the model is the same either for whole-cell or for patch clamps. Component

values given in the figure are typical for a medium size whole-cell experiment. The input capacitance, C_{in}, is contributed by the physical characteristics of the glass electrode. Series resistance R_s is a sum of electrode tip resistance, resistance between the tip of the electrode and the intracellular fluid and between the membrane and the ground electrode.

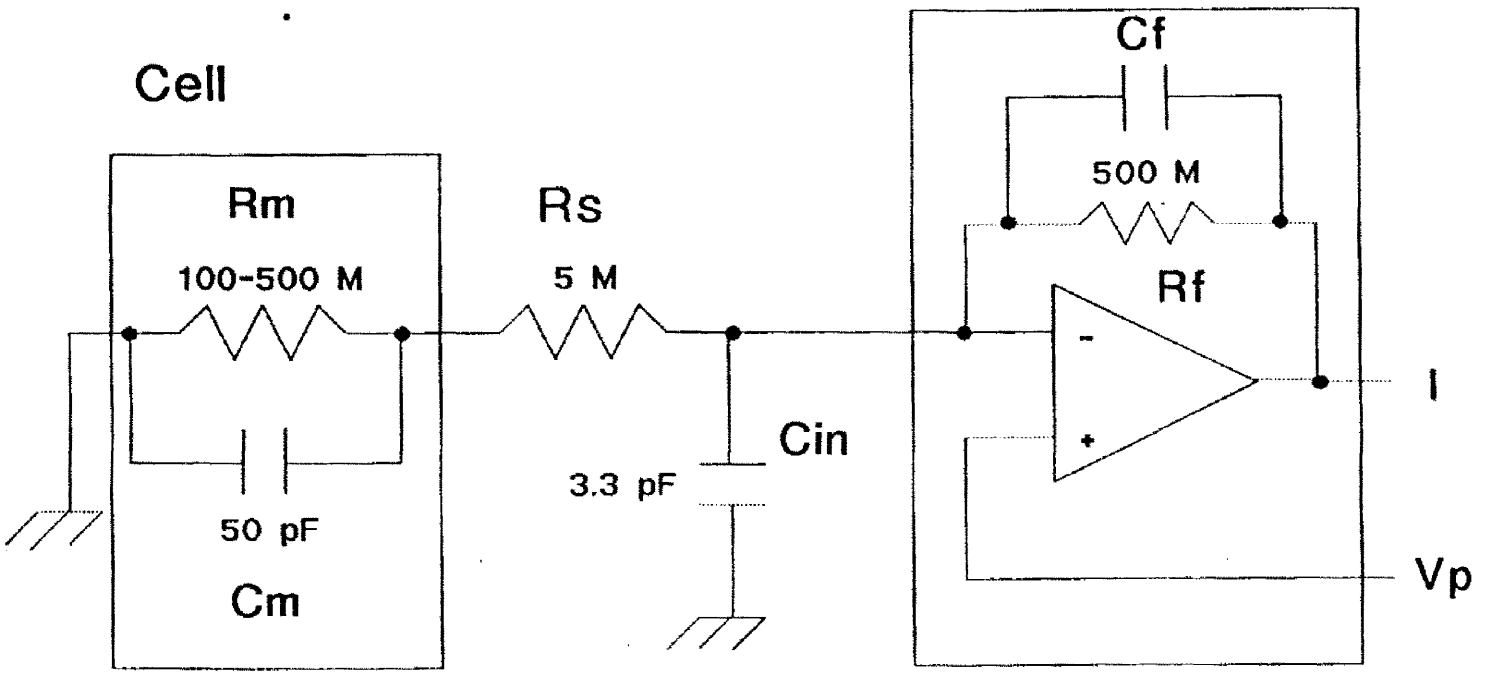


Figure 3.1B: Lumped-parameter circuit equivalent to the setup in Figure 3.1A. The feedback resistor, R_f, is given as 500 megohms. Voltage $V=I_pR$ is proportional to the current required to maintain the potential across the membrane at V_{ref}. C_f is stray capacitance associated with the feedback resistor, and R_s and C_{in} are parameters attributable to the glass microelectrode.

For cardiac cell experiments the electrode tip resistance is by far the largest contributor to series resistance. The parallel combination of R_m and C_m is a lumped-parameter representation of the electrical

characteristics of the cell membrane. Unit area capacitance for lipid bilayers is typically on the order of 1 uf/cm^2. Measurements of membrane current in this environment are subject to errors and limitations from several sources. Much of the following is based on an analysis by Colquhoun and Hawkes (1983).

Bandwidth limitation: In theory the operational amplifier responds infinitely quickly to differences in potential between the positive and negative inputs. Practical op amps have slew rates on the order of 10^{-7} seconds, providing a maximum effective bandwidth of about 1.6 mHz. The feedback resistor, by the nature of its construction, has a small stray parallel capacitance. Parallel combination of resistance and capacitance acts as a low pass filter, limiting the response speed and therefore the bandwidth of the clamping circuit. Although high-quality feedback resistors are designed to minimize stray capacitance, for a typical commercial film resistor of 500 megohms and an effective stray capacitance of 50 fF, the bandwidth of the clamp circuit is limited to 6.4 kHz. The combination of feedback resistance-capacitance and finite op amp slew rate produces a second order low pass filter dominated by the effect of the feedback circuit. The bandwidth of the circuit is important since events which occur faster than the amplifier can respond are sharply attenuated or lost altogether.

Series resistance: Feedback current flowing across the membrane also causes a voltage drop across the series resistance equal to the product of the feedback current and R_S. The potential across the membrane differs from the desired clamp potential by an error factor, the magnitude of which is proportional to the magnitude of R_S relative

to R_m. This error may become particularly significant (up to 50%) during depolarizations of squid giant axons, since membrane resistance is lowest at that time (Binstock et al 1975). For the model in Figure 3.1b, the potential at which the membrane is clamped differs from the command potential by 1-5%.

Capacitive transients: The input and membrane capacitances create current and voltage spikes during changes in clamping potential as they charge from and discharge into the clamp circuit. Size and duration of the spikes depend on the magnitude of the capacitances and the magnitude of resistances in series or parallel. The time course of the transients may be very complex due to the presence of several distinct capacitances in differing locations.

ANALOG TECHNIQUES

Several different circuits and methods have been devised over the years to correct or compensate for the major sources of error. The bandwidth limitation imposed by the op amp and feedback resistor has characteristics which are determined by the design and physical construction of the feedback amplifier and which are therefore constant for each device. For the circuit described above the frequency response of the clamp circuit has a first order rolloff corresponding to the time constant R_fC_f, and a second rolloff at the corner frequency of the op amp.

The bandwidth of the amplifier can be extended by a correction circuit such as that shown in Figure 3.2. This circuit has a first

order rollup at the same corner frequency as the R_fC_f rolloff, and can extend the bandwidth up to the range of 40-100 kHz. Commercially available voltage clamp devices such as the Axopatch-1A (Axon Instruments) include a "boost box" which is adjusted to compensate for feedback amplifier bandwidth limitations. Errors from this source are effectively eliminated when the adjustment is carried out properly.

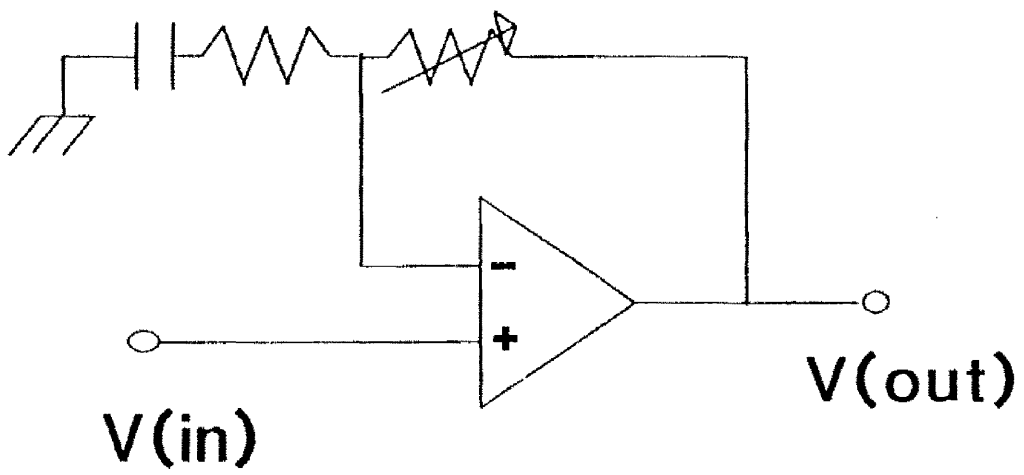


Figure 3.2: A "negative capacitance" amplifier circuit used to compensate for the low pass filter effect of R_fC_f. The variable resistor is adjusted to set the follup corner frequency of this circuit to the same frequency as R_fC_f.

The time course of current recordings from voltage clamp experiments is altered by the combined effects of series resistance and capacitive elements. Hodgkin, Huxley and Katz (1952) used careful electrode placement and modifications to the feedback circuitry to eliminate much of the series resistance effect, typically attaining an estimated 60% error correction. More recently, Sigworth (1980) used a

second order low-pass Bessel filter in the feedback loop to stabilize the response of the clamp circuit. Moore et al (1984) have developed an "active bridge" circuit designed to provide error correction approaching 100%. A method based on numerical analysis techniques has been devised by Palti and Cohen-Armon (1982) for off-line correction of series resistance errors. As an alternative to feedback circuitry modifications some investigators, noting that series resistance error is proportional to the magnitude of membrane current, have limited membrane current by lowering the concentration of sodium in the extracellular fluid (Moore et al, 1984). While this method does reduce sodium currents it also reduces the strength of the recorded signal and may significantly lower the signal-to-noise ratio and the physiological effects of lowering the extracellular sodium concentration may alter the results of the experiment.

Series resistance tends to dampen capacitive transients and efforts to compensate for series resistance errors with active analog circuitry are complicated by the fact that as the level of compensation increases, the capacitive components begin to destabilize the feedback amplifier. At levels much above 70% the clamp circuitry exhibits ringing and oscillation. Additional circuitry, usually involving "negative capacitance" amplifiers, is frequently used to cancel capacitive transients by injecting current into the clamp circuit in opposition to the currents created by the capacitances. This circuitry differs from that described to compensate for operational amplifier feedback resistor characteristics mainly in that the time constant for the amplifier compensation circuit is usually fixed at the time of

manufacture and requires little adjustment. Capacitive transients attributable to experimental conditions must of necessity differ between experiments, sometimes widely so, and the compensation circuitry should therefore be very flexible. Note that the series resistance and capacitive transient circuits must be adjusted together, that is, interactively, to minimize distortions in the voltage clamp recordings.

To illustrate how all of this works in the experimental environment, in the following example experiment a whole-cell clamp recording was made from canine ventricular myocytes using an Axopatch-1A. The procedure is paraphrased from the Axopatch-1A Patch Clamp Theory and Operation manual by Finkel (1986), with permission.

The Axopatch has four controls for input capacitance compensation (two each for fast and slow components), three for series resistance, and one for membrane capacitance. A glass electrode of 3-6 megohms tip resistance is brought into contact with the cell membrane with micromanipulators and a high-resistance seal forms between the lumen of the electrode and the membrane. A repetitive square wave is generated with an oscillator internal to the Axopatch, and the fast time constant (tau) and magnitude controls are adjusted to minimize the fast transients. Occasionally after the fast adjustment has been completed a slow transient is observed, and this is minimized with the slow tau and magnitude controls. After the fast and slow adjustments are finished the patch of membrane enclosed by the lumen of the electrode is ruptured either by vacuum or by electrical impulse, and a new set of transients appears. The series resistance and whole-cell capacitance

controls are adjusted to minimize the additional transients, and the fast and slow tau controls can now be retuned if necessary. Note that the series resistance must be substantially smaller than the membrane resistance for accurate measurement of and compensation for series resistance and whole-cell capacitance.

DIGITAL TECHNIQUES

Laboratory microcomputers employed for data collection and analysis make possible another approach to correction and enhancement of voltage clamp records. The combination of cell, electrode and recording device may be viewed as a system acting upon the signal of interest, in this case the membrane currents, to produce the output signal observed by the experimenter. From this perspective, the equation describing the relationship of the observed output to the input and the system is

$$y(t) = x(t) * h(t)$$

where y(t) is the observed output, x(t) is the signal of interest and h(t) is the "impulse response" of the system. The " * " symbol represents the convolution in the time domain of the input, x(t), with h(t). This system is shown schematically in Figure 3.3. To explain how this approach is used, we will begin with a simple case where the system acts as a low pass filter with cutoff frequency f_c. Input signals composed of frequencies below f_c are passed through the system without distortion, but higher frequencies are increasingly attenuated. A system of this type is often depicted in a log/log plot known as a

Bode magnitude plot (Figure 3.4a). The ordinate is the base 10 log of frequency, either in radians per second or Hertz and the abscissa is in decibels defined as $20\log_{10}$(output/input). This type of plot, which is also called a "frequency response curve", is very useful in that the cutoff frequency or frequencies of the system may be estimated by inspection and, conversely, if the equation describing the system is known the Bode plot can be sketched directly from the coefficients. With an ideal first order filter the degree of attenuation of higher frequencies increases by 20 db/decade of frequency. Higher order filters have correspondingly faster rates of frequency attenuation.

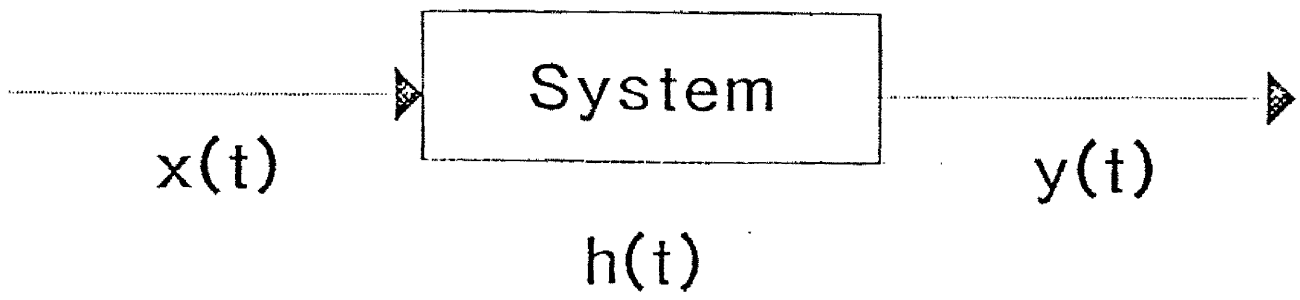


Figure 3.3: Generalized diagram of a single system acting on an input signal, x(t), to produce the output signal, y(t), by time domain convolution with h(t), the system impulse response.

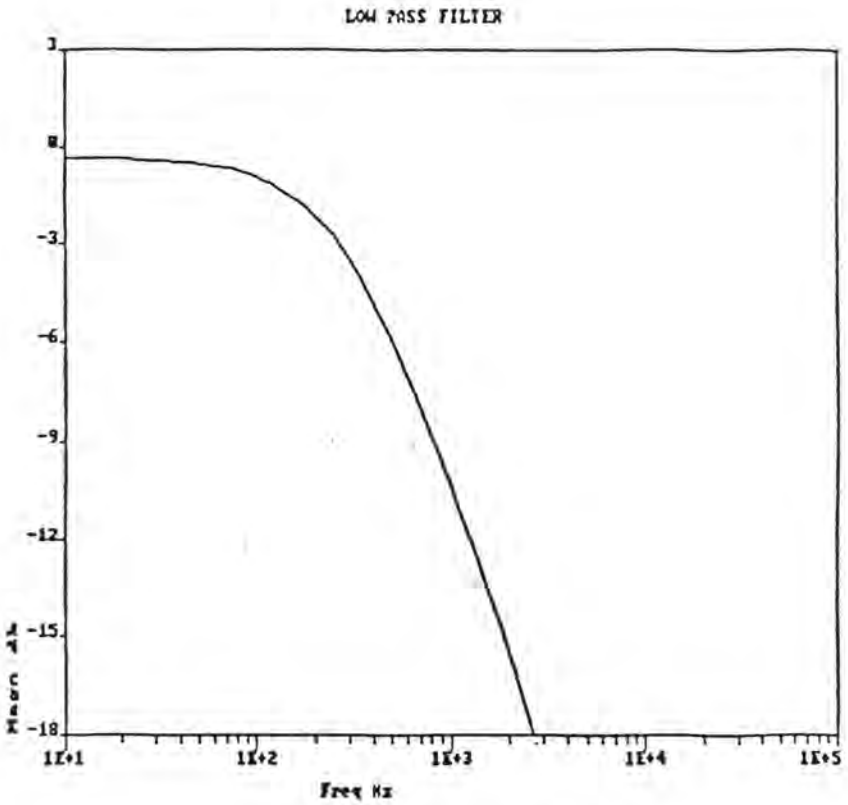


Figure 3.4A: Low pass filter.

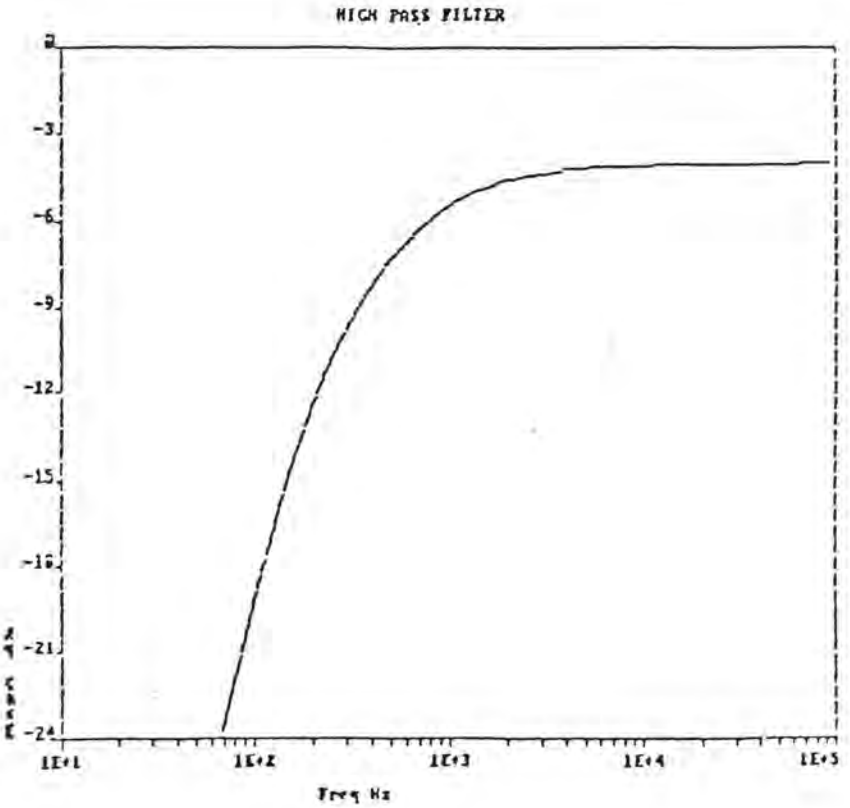


Figure 3.4B: High pass filter.

Figure 3.4: Bode magnitude plots of a first order low pass filter, Panel A, and a first order high pass filter, Panel B. Corner frequency for both filters is 340 Hz.

The Laplace transform frequency domain representation of a first order low pass filter is

$$Y(s) = (1 / ((s/p_i)+1))X(s)$$

where p_i is the system "pole" corresponding to the corner frequency of the filter. (The designation "first order" refers to the single system pole.) This equation has the form

$$Y(s) = H(s)X(s)$$

where $H(s) = (1 / ((s/p_i)+1))$ is the system transfer function impulse response. The time domain representation, h(t), is found by taking the inverse Laplace transform of H(s) which yields

$$h(t) = p_i * \exp(-p_i * t).$$

In the discrete case, i.e. for data points in the microcomputer, the continuous signal from the recorder is sampled at regularly spaced points "T" seconds apart. The impulse response at the "k^{th}" sample point is

$$h(kT) = \int_{(k-1)T}^{kT} (p_i)(\exp(-p_i * w))\, dw$$

$$= \exp(-p_i * kT) - \exp(-p_i * (k-1)T).$$

Finally, the output signal at a discrete point in time, y(kT), is computed from the convolution summation as

$$y(kT) = \sum_{i=0}^{k-1} h(k-i) * x(i)$$

Another simple case is the system which acts as a first order high pass

filter, in which frequencies progressively below the cutoff point are progressively attenuated at the first order rate of 20db/decade. The Bode magnitude plot is shown in Figure 3.4b and the transfer function representation is

$$H(s) = s / ((s/p_i)+1).$$

This function is referred to as "improper" or "feedforward" since the order of the numerator and denominator is the same, the term "feedforward" referring to the fact that an input has a corresponding instantaneous output as opposed to a slight delay between input and output for the low pass filter. Converting the transfer function to the time domain and integrating to solve for the discrete case yields

$$h(kT) = 1 + \exp(-p_i * kT) - \exp(-p_i * (k-1)T)$$

and the output, y(kT), is found by the convolution summation given earlier.

Low and high pass filters can be combined in the same system to produce a bandpass filter. A typical frequency response curve, shown in Figure 3.5a, illustrates the action of the filter in passing without distortion only those frequencies between the high pass and low pass cutoffs. This effect can be achieved by sending the input signal through one of the filters, then passing the output of that filter through the other. Note that the order of filtering has no effect on the form of the output. Figure 3.5b is a system schematic of bandpass filtering. In the frequency domain the transfer function of the bandpass filter is the product of the transfer functions of the high and low pass filters:

$$H(s) = (H_{lo}(s))(H_{hi}(s)).$$

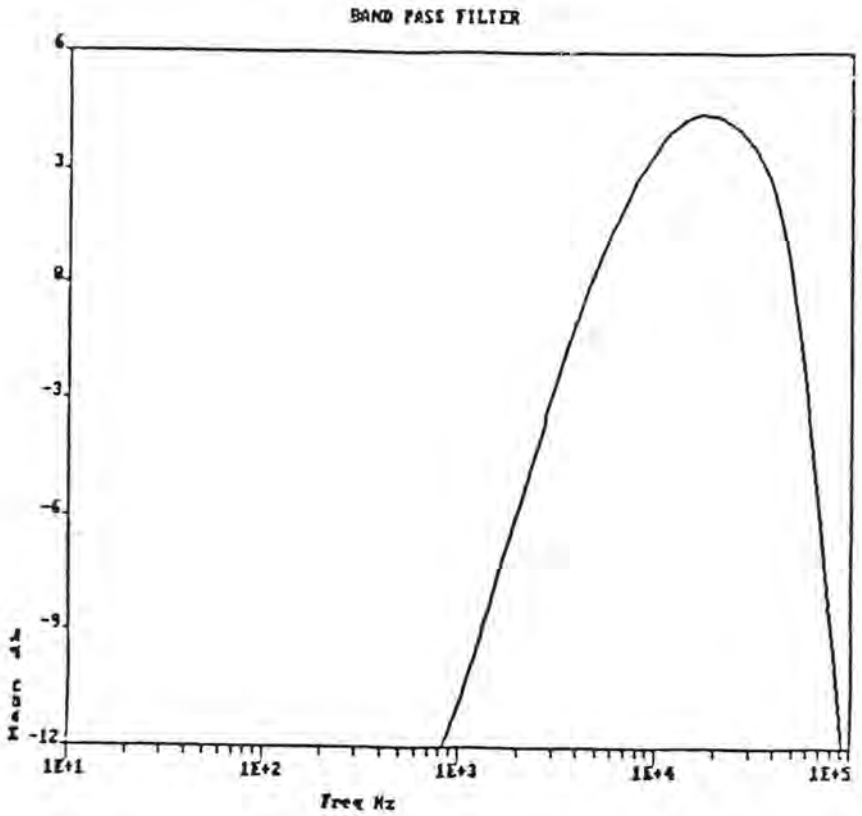
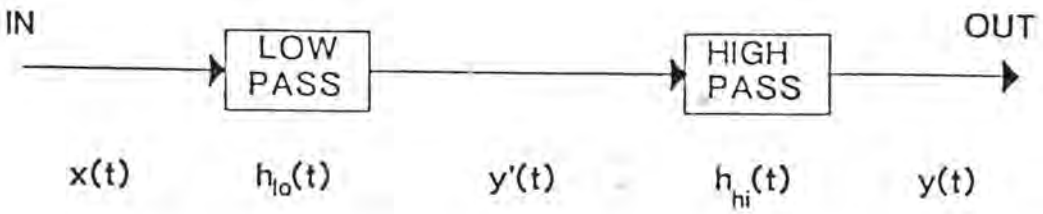


Figure 3.5A: Band pass filter.



$$y'(t) = x(t) * h_{lo}(t)$$

$$y(t) = y'(t) * h_{hi}(t)$$

Figure 3.5B.

Figure 3.5: Bode magnitude plot of a band pass filter showing attenuation of frequencies above and below the band cutoff points. Panel B shows how a band pass filter can be formed by series connection of a low pass and a high pass filter.

Extending this line of reasoning it is seen that very complex systems often can be broken down into a series of simpler functions, ordered arbitrarily, through which a signal is passed seriatim.

Now suppose that our recording system acts as a bandpass filter. The signal passing through the system is distorted in that very rapid events will be attenuated, as will very slow changes in the amplitude of the signal. The severity of the distortion is a function of how much of the signal energy lies above the low pass cutoff or below the high pass cutoff. Much of this signal energy can be recovered by artificially extending the high and low pass cutoff points and this is exactly what the capacitance and series resistance compensation circuits are designed to do. This is also the same operation discussed earlier with regard to extending the bandwidth of the feedback amplifier used as a voltage clamp. The same effect is achievable mathematically and, therefore, in the computer, in the following manner.

The bandpass filter is two simple filters, low and high pass, in series. We can add two more blocks to the system, one of which is the inverse of the low pass, and one for inverse high pass. Recalling that in the frequency domain the transfer function of two systems in series is the product of the transfer functions of each of the systems, we can define the inverse filtering functions as

$$H_{inv}(s) = ((s/p_i)+1) \; / \; ((s/p_2)+1)$$

where $p_2>p_i$ for inverse low pass, and $p_2<p_i$ for inverse high pass. The net effect of passing a signal through a filter, then the inverse of the filter, is to substitute the higher or lower cutoff frequency. This

substitution is required because, analogous to the case where very good series resistance compensation causes instability, perfect compensation can cause unpredictable results. The analog and digital implementations of this scheme are mathematically identical and the concept can be extended as before to more complex, higher order systems.

Digital signal compensation can now be perceived as a two step operation in which first the characteristics of the system, i.e. the form and coefficients of the transfer function are determined, and then corrective measures are employed to extend the bandwidth of the system as appropriate. With the Axopatch analog circuitry both steps are taken simultaneously. Adjusting the test signal with the compensation controls is equivalent to determining the system parameters, and, once set, compensation occurs continuously at those settings. Although it may be possible to use the control settings to determine the parameters for digital signal compensation as well, the actual analog settings in each experiment depend to some extent on the ability and experience of the experimenter. We have observed some variability in settings between operators and even within the same operator using the recorder at different times.

SYSTEM IDENTIFICATION

There is a number of different techniques for system identification, three of which have been tested extensively and are described below.

STEP RESPONSE

The square wave used to set the analog compensation controls is a

series of step functions. Theoretically, the response of a system to a step function is a series of summations of the impulse response terms:

$$r(kT) = \sum_{i=0}^{k} h(iT).$$

From this relation, the impulse response terms can be computed as

$$h(kT) = r(kT) - r((k-1)T).$$

In attempting to use this approach two major drawbacks were encountered, one general and one specific to our hardware. Usually the response curve recorded as the system output is contaminated to varying degrees by noise. This can be overcome to some extent by averaging a number of step responses, but when the h(kT) terms become numerically small the variance of the terms becomes correspondingly and often unacceptably large. For rapidly responding systems this effect may completely obscure the impulse response terms. Secondly, our A/D board is not capable of simultaneous A/D and D/A conversions, and for this reason we have designed the peripheral processor described in the previous chapter to pass commands to the Axopatch while the computer is taking data. Using the PPU, data acquisition can be timed to coincide with the onset of the step waveform but there is always some jitter between the time the A/D board receives the command to acquire data and the time at which data acquisition actually begins. At fast sample rates the jitter reduces the time correlation between samples of successive steps and tends to blur the calculated impulse response coefficients.

Even when sufficiently accurate recordings are obtained to give a good estimate of the impulse response terms, analysis of anything but a

very simple system is difficult. A good model of the system, including a priori information about the orders of the terms, must be available and the ranges of the coefficients of the model must be narrow and clearly defined so that curve fitting of the impulse response terms can produce good estimates of the coefficients. Our experience is that this relatively ideal situation is not often encountered.

WHITE NOISE

The continuous curve depicted in the Bode magnitude plot has a magnitude point at every frequency along the ordinate. If a signal which contains equal energy at every frequency in the range of interest is applied to the input of a system under study, measurement of the response of the system can be made simultaneously at each point on the frequency scale. This would supply the information for the plot and thus serve to characterize the system. Such a signal is termed "white noise", and is the product of a completely random or Gaussian process. Fluctuations in white noise signal amplitude are distributed normally with mean signal level of zero and variance proportional to signal power. To characterize a system using white noise requires an electronic source of noise which has equal energy in all frequencies in the range of interest. Because the noise must only be "white" in a certain frequency band, this signal is sometimes called "band limited white noise". After applying the signal to the system input, Fourier transform analysis is performed on the recorded, digitized output signal to measure the response of the system at each of a series of equally spaced frequencies. The actual frequencies available for analysis depend on the time interval between digitization points

(samples) and the number of samples taken. Since by definition the white noise input signal has equal energy at all frequencies this technique is theoretically equivalent to applying sine wave inputs to the system at each of the equally spaced frequencies simultaneously. The magnitude of the Fourier transform of the recorded signal is an estimate of the frequency response of the system. In most cases the Fourier transforms of a number of recordings are averaged to improve the estimate.

A good source of band limited white noise for this application is not easily obtainable. The least expensive commercial noise generator we could find was several thousand dollars, more than we felt was practical for this application. Attempts to construct analog noise generators based on reverse biased diodes were not entirely successful, and it was determined that digital pseudorandom noise generators were too complex and time consuming to build. However, the system itself generates noise which we felt might be used to determine the feasibility of this approach. This noise comes from several sources, including junction or "shot" noise in the semiconductor elements in the electronics and Johnson or thermal noise in the resistive elements. Since the level of thermal noise is proportional to electrical resistance, the very high resistances presented by the cell membrane and glass electrode tip might generate sufficient noise for the analysis. Several noise recordings were made with the Axopatch electrometer set for very high gain and estimates of the power spectra were computed using Welch's modification of the Bartlett method (Oppenheim, Shafer 1975). A typical spectrum, plotted on normal scales,

is shown in Figure 3.6. One surprising result was the appearance of peaks centered on harmonics of 1 kHz. The source of this signal has not been found but we speculate that some device in the neighborhood of the lab generates a very weak signal not observable at normal levels of amplification. Also, since despite our best efforts we were not able to eliminate this artifact completely, it is possible that the Axopatch itself is generating the signal, perhaps in the feedback circuit in the headstage.

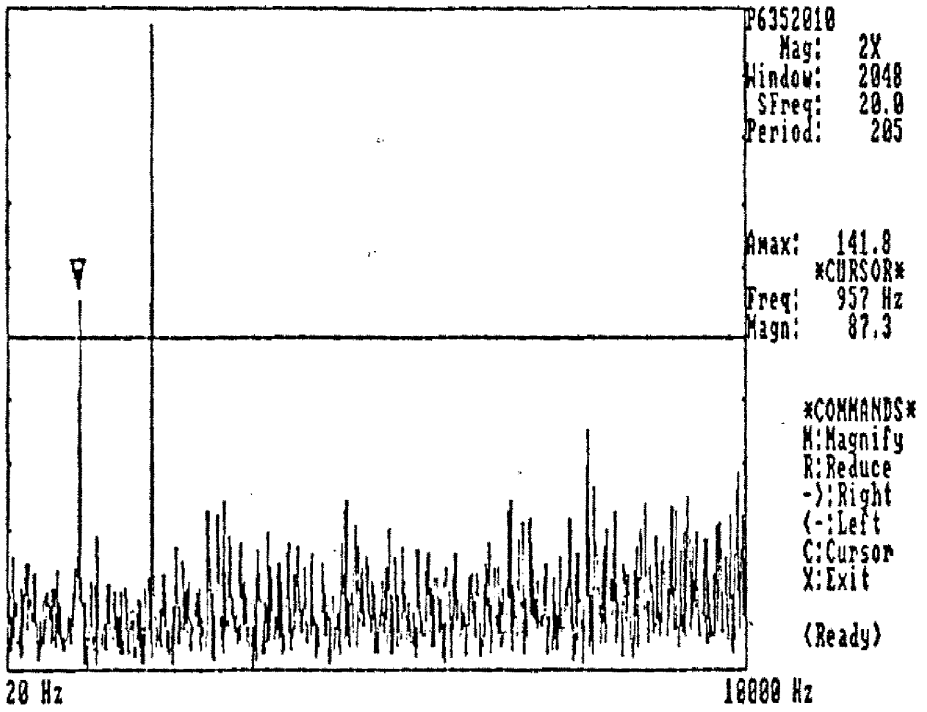


Figure 3.6: This is the average of the Fourier transform magnitudes of ten noise recordings taken from the Axopatch-1A with a ten megohm resistor connected from the headstage to ground. Note the peaks at approximately 1 kHz and 2 kHz possibly attributable to external interference.

When the power spectrum is plotted on the Bode log-log scale another difficulty becomes evident. Discrete Fourier power spectra are estimates of the actual spectra in that only a finite amount of data are available for analysis. The underlying process is theoretically of infinite duration and so requires an infinite amount of data to specify exactly. In practice, taking a finite data run has the effect of "windowing" the data by multiplying a continuous infinite data stream with a window of finite duration. Truncating the data in this manner causes ripples in the frequency domain. These ripples are, in effect, the Fourier transform of the window convolved with the transform of the data. This effect is moderated in the Welch method by specifying a smoothly curving window that reduces the magnitude of the data points near the ends of the data run.

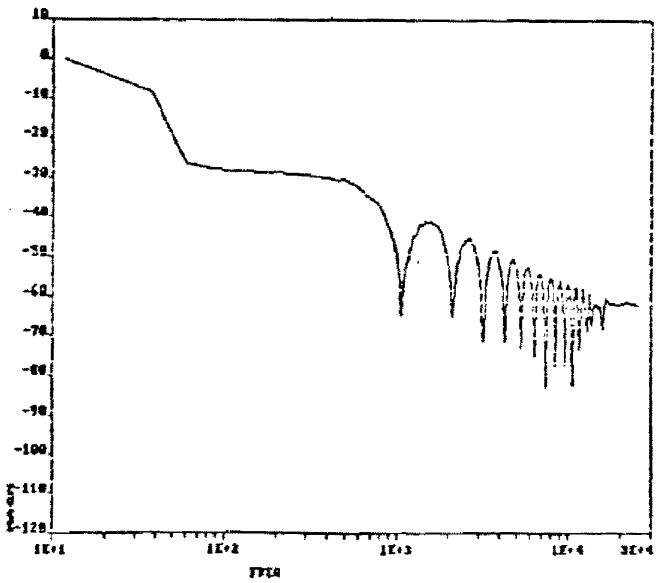


Figure 3.7: Log-log plot of the Fourier transform magnitude of a data file consisting of the coefficients of a Hamming window. The magnitude curves above 1 kHz represent side lobe energy which appears as "ripples" in frequency domain representations of data files processed with the Hamming window.

Figure 3.7 is a Bode plot of the Fourier transform magnitude spectrum of the Hamming window used in these experiments. It is apparent that the signal strength of the internal noise is so low that, as illustrated in Figure 3.8, the transform of the Hamming window almost completely dominates the transform of the noise data. It is possible to compute the transform of the window and, in theory, remove it from the response curve but in practice it is difficult to specify the transform with sufficient precision to eliminate the window effect without severely distorting the data, especially when the form of the transform of the data is not known a priori. In addition, the computational burden increases along with increasing manipulation of the data, with a concomitant requirement for greater operator participation in the

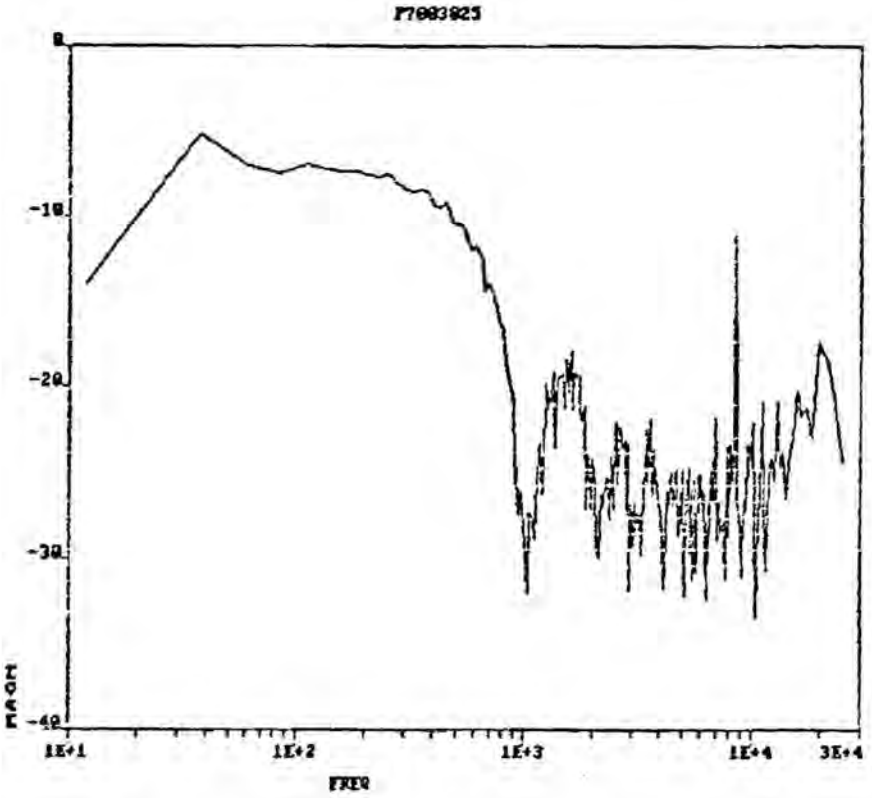


Figure 3.8: Bode magnitude plot of a noise recording taken at the same time as those in Figure 3.6, showing side lobe peaks characteristic of the Hamming window. The window effect almost completely masks the frequency response attributable to the noise. Note the peak at approximately 9 kHz, a harmonic of the 1 kHz contamination discussed in the text.

Ultimately we were able to borrow a noise generator, General Radio type 1390-B, which produced signal levels high enough to permit application of the white noise analysis approach. A series of test spectra run directly on the output of the generator, shown superimposed in Figure 3.9A, digitized at three different sampling frequencies, indicates that the noise spectrum was sufficiently flat within our range of interest. The presence of spikes at 60Hz and 120Hz, indicating a weak filter capacitor in the generator power supply, did not interfere with the analysis.

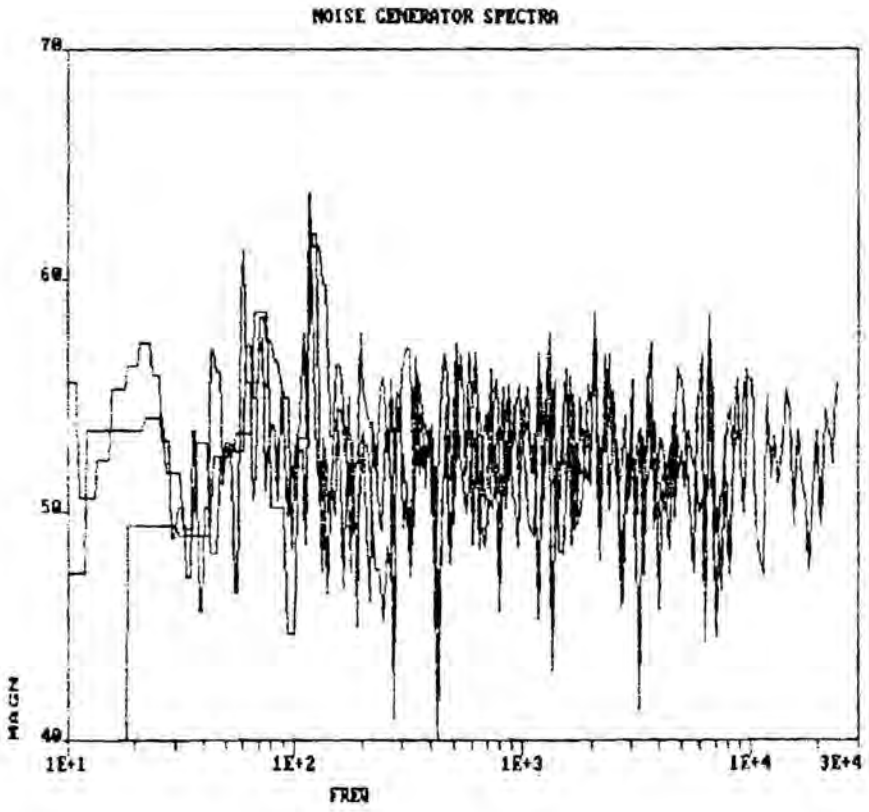


Figure 3.9A: Three superimposed noise spectra from recordings made directly from the noise generator output. Recordings were digitized at 10 kHz, 20 kHz and 50 kHz to eliminate sampling effects. Note peaks at 60 Hz and 120 Hz probably due to a weak filter capacitor in the generator power supply.

After measuring the output of the generator, the Axopatch was set up with a 10 megohm resistor as a dummy cell and another series of spectra (Figure 3.9B) was made to ensure that the electrometer/recording system had no significant effect on the noise spectrum.

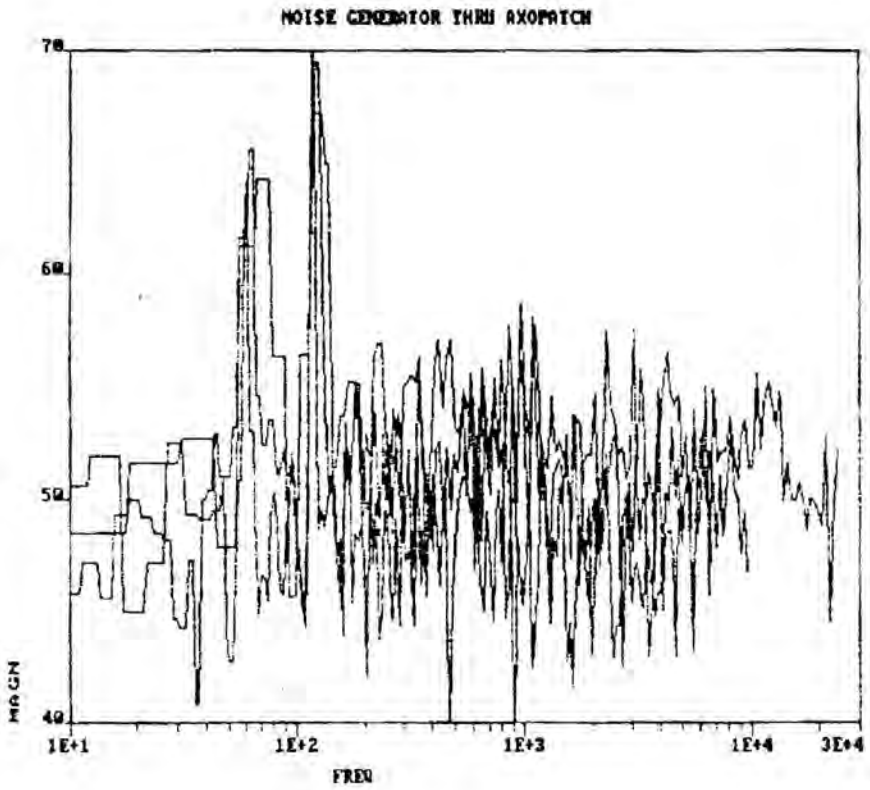


Figure 3.9B: Three superimposed noise spectra recorded from the output of the Axopatch electrometer, using the noise generator. These recordings were taken at the same sampling rates as in Figure 3.9A and also have peaks at 60 Hz and 120 Hz.

Lastly, a set of calibration spectra was taken using the Axopatch internal four-pole Bessel filter at various settings and also using analog capacitor-resistor high pass filters. Three representative curves, Figures 3.10a-c, show that the method as implemented does

indeed produce system transfer response curves within the sample rate limitations of the A/D board.

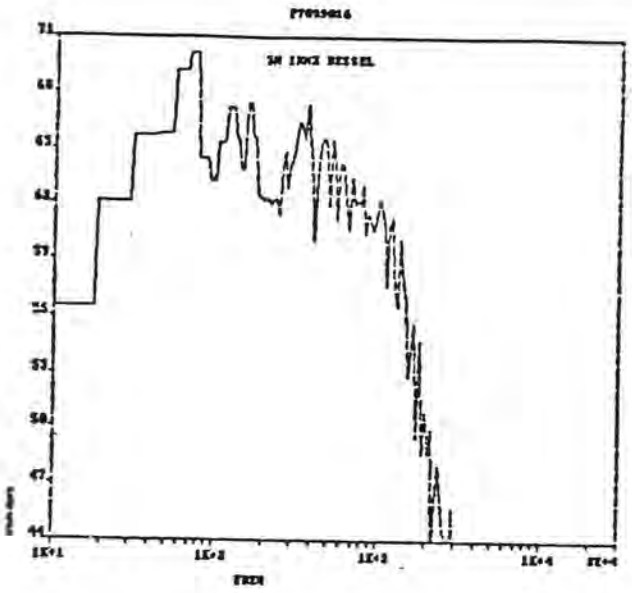


Figure 3.10A: Low pass filter, f_c=1 kHz.

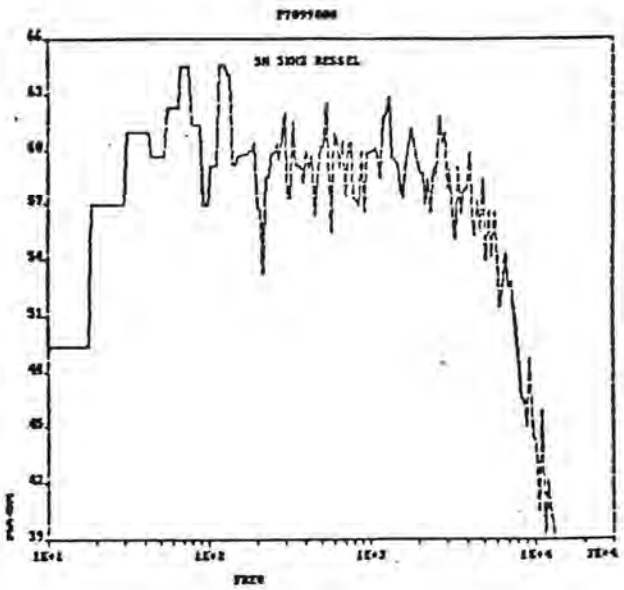


Figure 3.10B: Low pass filter, f_c=5 kHz.

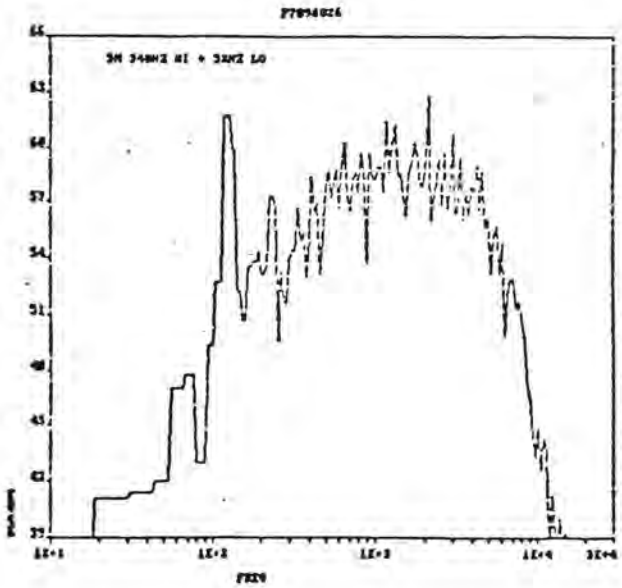


Figure 3.10C: Band pass filter, $f_{c,lo}$=340Hz, $f_{c,hi}$=5kHz. These three spectra, taken with the noise generator connected to the Axopatch, demonstrate the utility of noise spectra in determining the form and estimating the coefficients of filter transfer functions.

FREQUENCY RESPONSE

The third major type of approach is also one of the oldest and most commonly used. To determine the frequency response of a system directly, a waveform generator is used to inject sine waves of fixed amplitude and specified single frequencies into the system. The ratios of the peak-to-peak amplitudes of the input and output signals are used to compute the magnitude of the response of the system at each frequency. Results are plotted on the log-log Bode scale in the usual fashion. Use of this method is so widespread that there are devices on the market, commonly known as frequency spectrum analyzers, which carry out this operation automatically. Frequency spectrum analyzers use a swept frequency generator to inject a signal of continuously varying frequency which repetitively "sweeps" the frequency range of interest. The system output signal is analyzed and the results are displayed on an oscilloscope or strip chart recorder as a Bode plot. Unfortunately for us, these machines tend to be quite expensive, on the order of many thousands of dollars. The method, however, has many advantages. Little or no a priori knowledge of the characteristics of the system is required. Since the response measurement technique uses a limited number of specified frequencies, problems which might arise in measuring one or several of the individual frequencies usually will not invalidate the results. All calculations are in the time domain and very few assumptions are required regarding the nature of the system or of the analysis process.

Early attempts to implement frequency response analysis involved using the microcomputer to digitize the input and output signals simul-

taneously, then compute and display the results. This method was severely hampered by the limited sampling rate of the A/D board, at that time 12 kHz per channel for two channels, which meant that reliable results could be obtained only up to a few kilohertz.

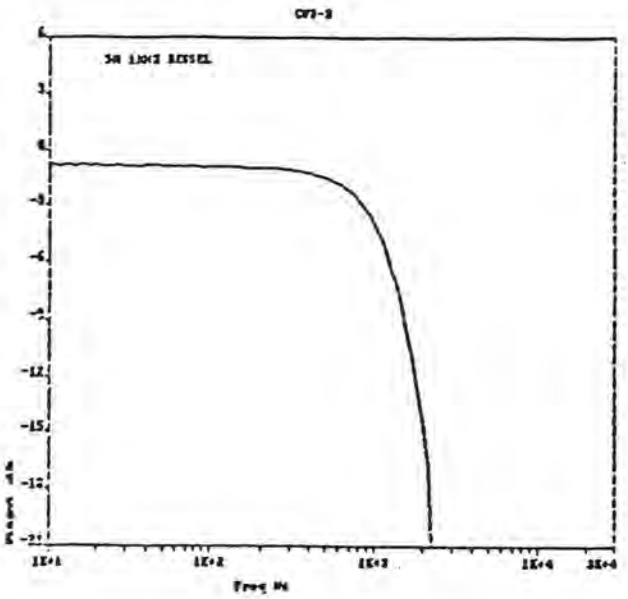


Figure 3.11A: Low pass filter, f_C=1 kHz.

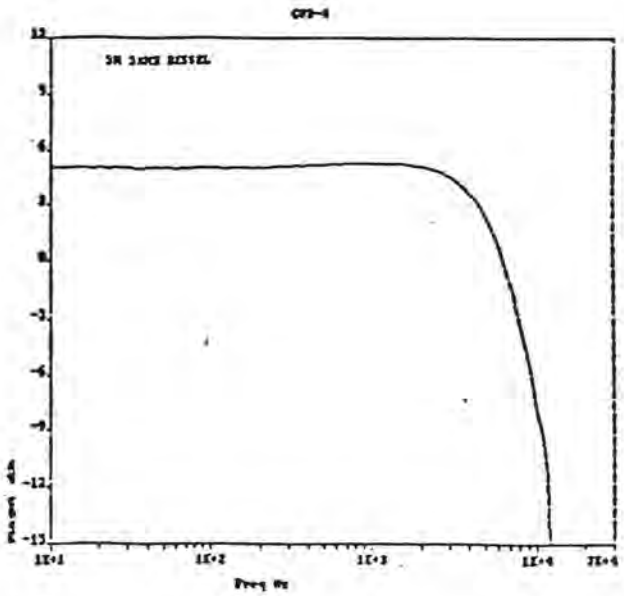


Figure 3.11B: Low pass filter, f_C=5 kHz.

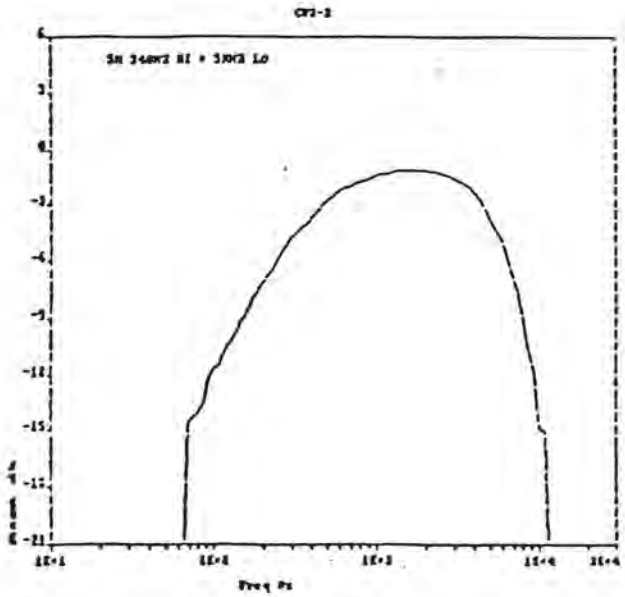


Figure 3.11C: Band pass filter, $f_{C,lo}$=340Hz, $f_{C,hi}$=5kHz. These three spectra were made with the frequency response technique, under the same conditions as Figures 3.10A-C.

We then designed and built a set of amplitude detectors which produces DC voltages proportional to the peak to peak amplitude of the input and output sine waves. This relieves the system of the constraints imposed by limited sampling rate since only the DC levels at each frequency need be digitized. Information regarding phase shifts imposed by the recording system is not provided by this technique.

This method was tested under conditions similar to those used in the white noise analysis and a set of typical response curves is shown in Figures 3.11a-c. These curves are sufficiently accurate to permit graphical determination of the system corner frequencies using the method discussed earlier. This experience leads to the conclusion that, in this application, the frequency response approach to system identification is the method of choice.

DIGITAL COMPENSATION

Once the characteristics of the system have been identified, digital compensation becomes a fairly straightforward computational problem. The impulse response of the system is determined from the response curve and, as discussed earlier, the convolution summation produces the output signal from the input and the system impulse response. A difficulty with the convolution summation method lies in the computational burden required to compute each data point in the output, particularly if the impulse response has many significant terms. The finite-difference equation model compresses the pulse response model into a very few coefficients of past inputs and outputs. In the literature, the discrete finite-difference equation is also called an autoregressive moving average (ARMA) model. This equation has

the form

$$y(k) = -a_{n-1}y(k-1) - a_{n-2}y(k-2) - \ldots - a_0y(k-n)$$
$$+b_{n-1}x(k-1) + b_{n-2}x(k-2) + \ldots + b_0x(k-n)$$

When the coefficients of the finite-difference equation are properly computed, this produces exactly the same output as the convolution summation with a fraction of the number of calculations. Only n number of past inputs and outputs are required for the finite-difference method. For the first order cases of inverse high and low pass filters, n = 1, so that only the previous input and output data points must be retained in memory to compute the present value of the output. The coefficients are determined uniquely by the first two terms of the impulse response as follows

$$b_0 = h(T)$$

$$a_0 = -h(2T)/h(T)$$

where, as before, T is the time interval between sampled data points. Including the improper feedforward term, the equation for inverse filtering is

$$y(kT) = (h(2T)/h(T)y((k-1)T) + (h(T)x((k-1)T) + x(kT)$$

The first two impulse response terms are computed from the transfer function discussed earlier. This technique produces the convolved output very quickly, with accuracy limited only by the accuracy of the system frequency response measurement.

METHODS

The technique used for system identification, based on the frequency response approach, is illustrated in Figure 3.12. The

Tektronix sine wave generator is connected to the head stage of the Axopatch and the outputs of the generator and the Axopatch are fed to identical amplitude detectors.

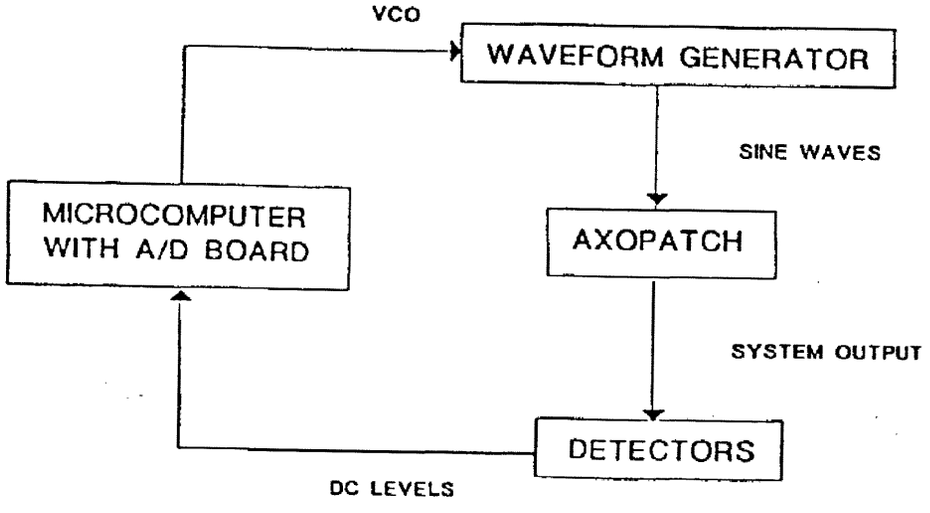


Figure 3.12: block diagram of the frequency response measurement system used to make the recordings shown in Figure 3.11. The A/D board produces a DC voltage used to control the frequency of the waveform generator through a voltage-controlled oscillator (VCO) input on the generator. Voltage levels from the detectors, proportional to sine wave peak to peak amplitude, are digitized by the A/D board to compute the frequency response points.

Circuitry for the detectors is depicted in Figure 3.13 and consists of a very simple high pass filter to remove DC drift and offset, followed by a halfwave detector to convert the AC sine wave to a corresponding DC level. The detectors were calibrated using sine waves of known amplitude in the range of 0.2-3.0V_{p-p}. The measured calibration points are shown in Figure 3.14. DC voltages from the detectors are sampled by the A/D board and used to compute the frequency response data points. The output of each detector is sampled ten times at 1ms intervals and the ten measurements are averaged. Two seconds later, another set of

measurements is taken and averaged. If the difference between the averages computed from the output signal detector is less than one percent, the last set of averaged detector levels is recorded and the signal generator is stepped to the next frequency. Repeated sampling and averaging and comparing the results of sequential averages ensures that the detector outputs have stabilized before the DC voltage levels are accepted.

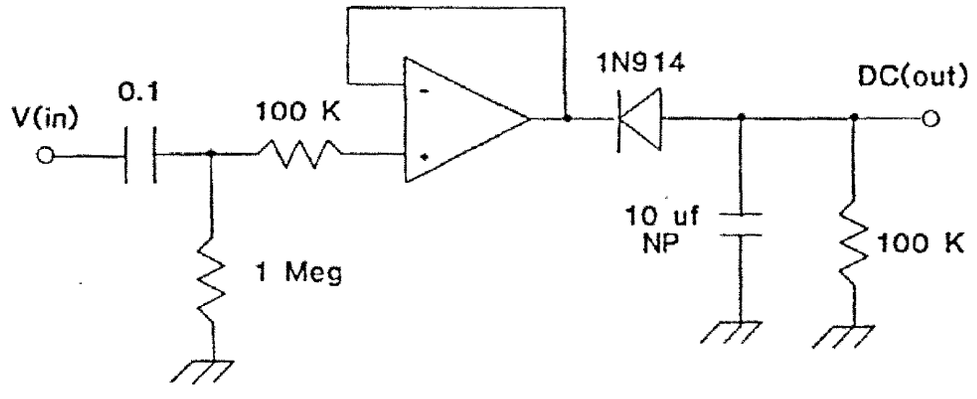


Figure 3.13: Analog sine wave peak to peak amplitude detector. Identical detectors are used to measure input and output waveform magnitudes. The input circuit to the op amp noninverting side serves to remove DC offset and drift, and the diode, capacitor and resistor convert the sine wave output of the op amp to a DC voltage which varies with varying amplitude.

Detector outputs falling between the calibration points, as most do, are interpolated using Newton's interpolary divided difference equation (Burden, Faires 1985). The frequency of the sine wave from the generator is controlled by a D/A port on the A/D board through a voltage-controlled oscillator input on the waveform generator. Calibration of the waveform generator as controlled by the D/A port was

checked with an Optoelectronics Model 8010 1GHz frequency counter. The frequency response measurement system as a whole was tested by measuring the response from high and low pass filters of known characteristics and this test setup, along with the results, is depicted in Figures 3.15a and 3.15b.

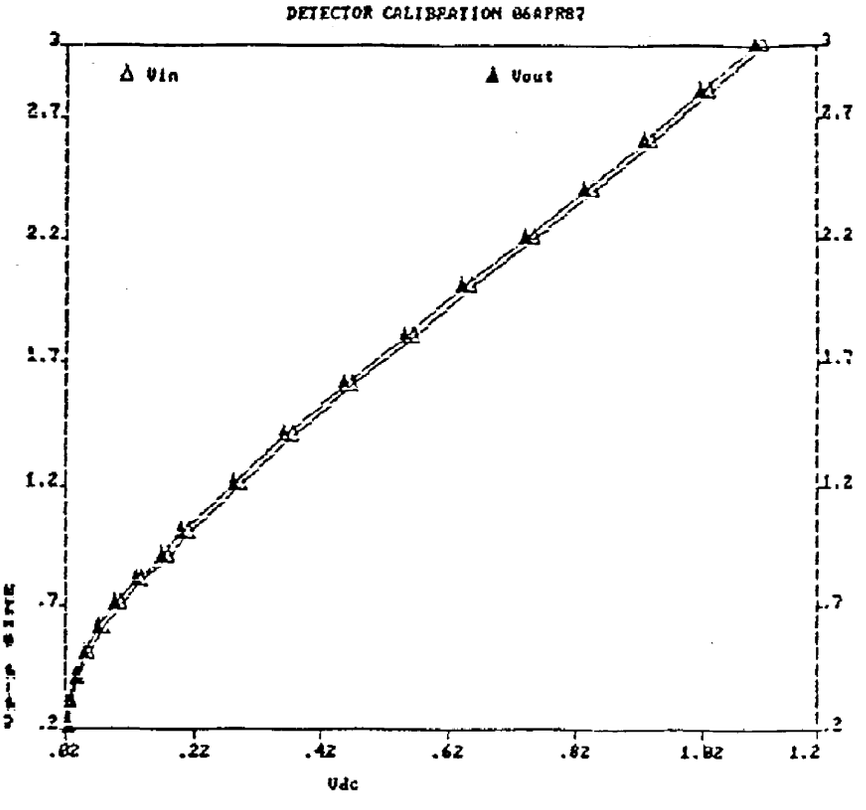


Figure 3.14: The analog amplitude detectors were calibrated using sine waves of known amplitude and the results are plotted in this figure. Curve fitting using a variety of functions failed to give good results, and detector outputs falling between calibration points are interpolated. Curvature at low amplitudes is due to diode nonlinearities at small signal levels.

The finite-difference equations of the inverse transfer functions have been programmed exactly as described. The software was tested and debugged on simulated data by running a series of digital filters and inverse filters on computer generated waveforms. Next, a series of

waveforms generated by the Tektronix waveform generator was passed through the analog low and high pass filters depicted in Figure 3.15 and inverse filtered to recover the original signals.

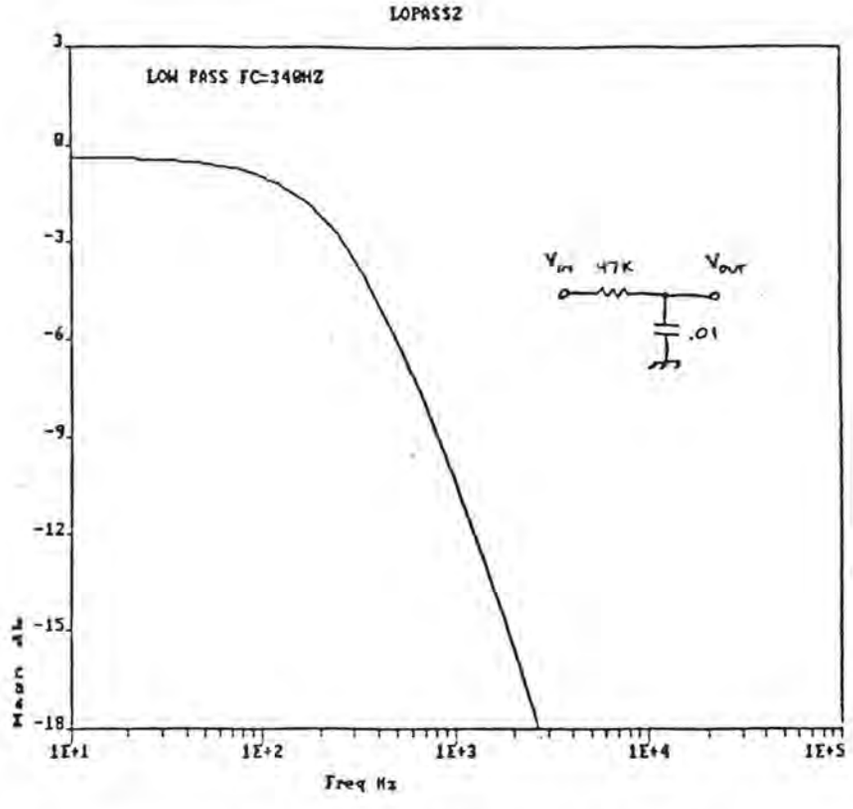


Figure 3.15A: Low pass filter calibration curve.

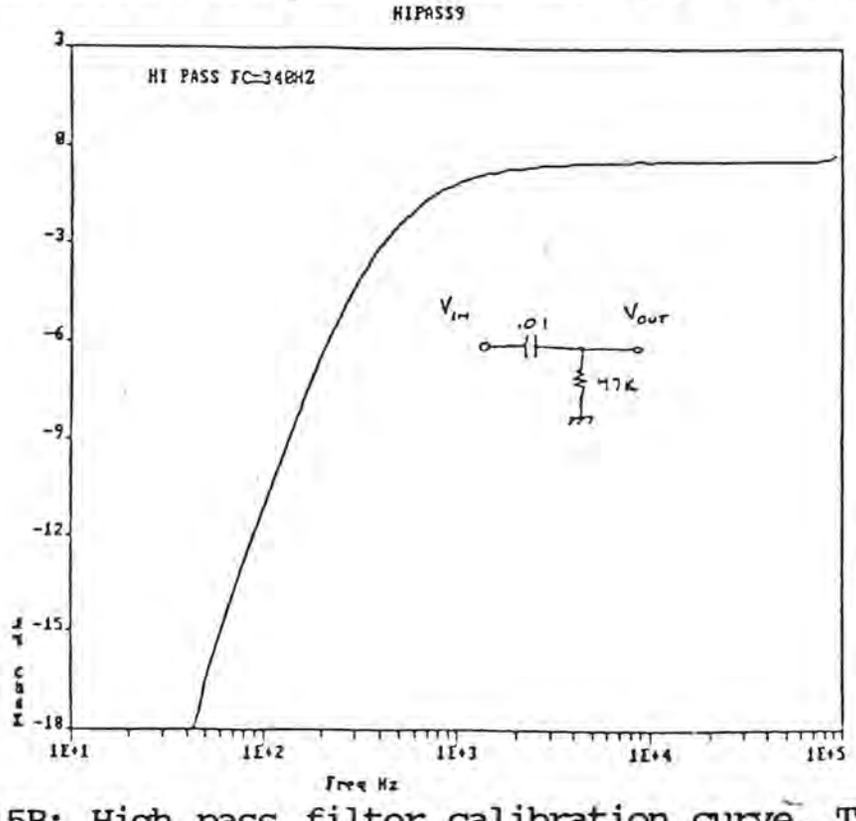


Figure 3.15B: High pass filter calibration curve. These two circuits were used to test and calibrate the frequency response measurement system.

Results of this series of tests are shown in Figure 3.16a-h. Figures 3.16a and 3.16c show 100Hz triangle and square waves, respectively, passed through the high pass filter with f_C=340Hz. Inverse high pass filtering with finite difference equations produced the corrected records of Figures 3.16b and 3.16d. Low pass filtering and inverse filtering are similarly shown in Figures 3.16e-h.

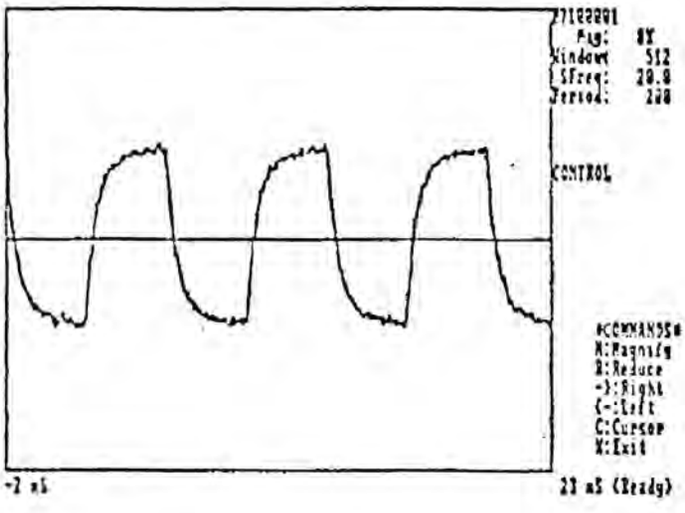


Figure 3.16A: Triangle waveform of 100 Hz, filtered by a high filter with f_C=340 Hz.

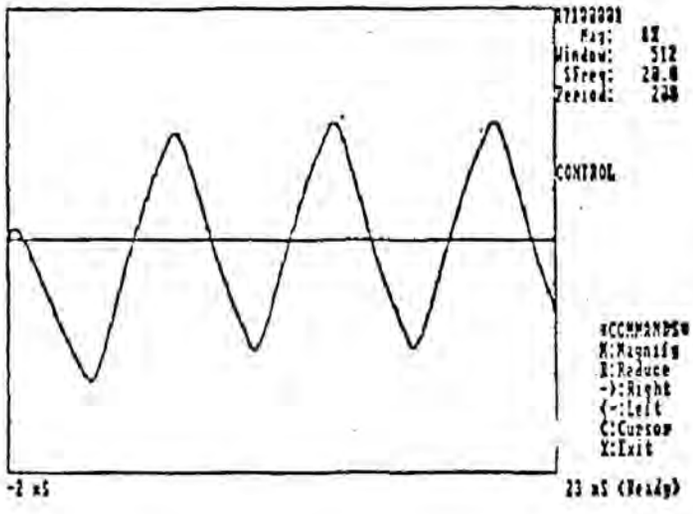


Figure 3.16B: Triangle waveform recovered by inverse filtering.

Another test was made with a glass electrode which had been attached to a cell in a bath as part of an unrelated experiment. The frequency response curve measured for this setup was used for the example of a bandpass filter curve which was shown in Figure 3.5a. A series of voltage clamp steps to elicit inward and outward currents was recorded from the cell and a typical data run is displayed as Figure

3.17a. Inverse filtering using parameters estimated from the response curve produced the waveform in Figure 3.17b. Since the high pass cutoff was very high, most of the data recovered was in the lower frequencies, hence the sharp reduction in high frequency noise.

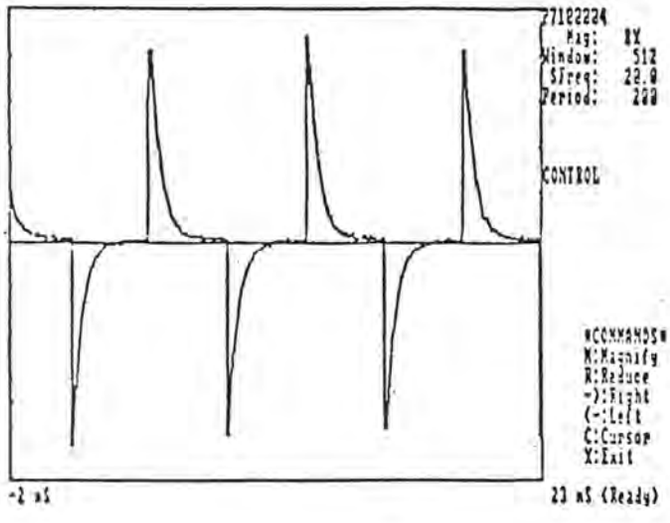


Figure 3.16C: Square waveform of 100 Hz, filtered by a high filter with f_C=340 Hz.

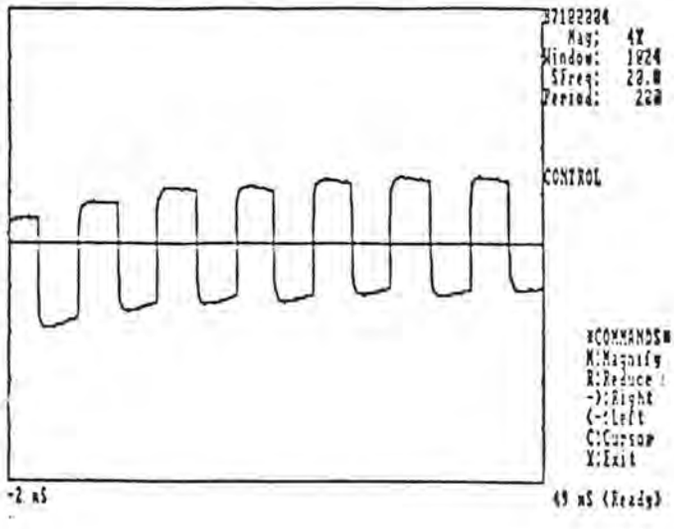


Figure 3.16D: Square waveform recovered by inverse filtering.

Inward currents elicited by the depolarizing steps and outward currents from the hyperpolarizing pulses are clearly distinguishable in the recovered waveforms.

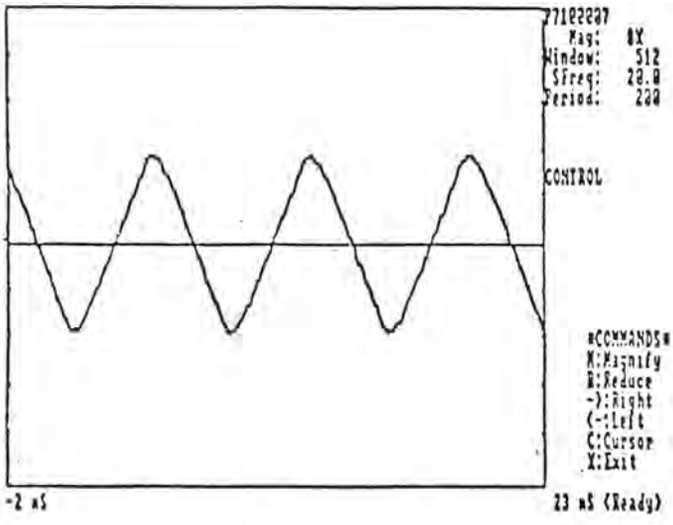


Figure 3.16E: Triangle waveform of 100 Hz, filtered by a low filter with f_C=340 Hz.

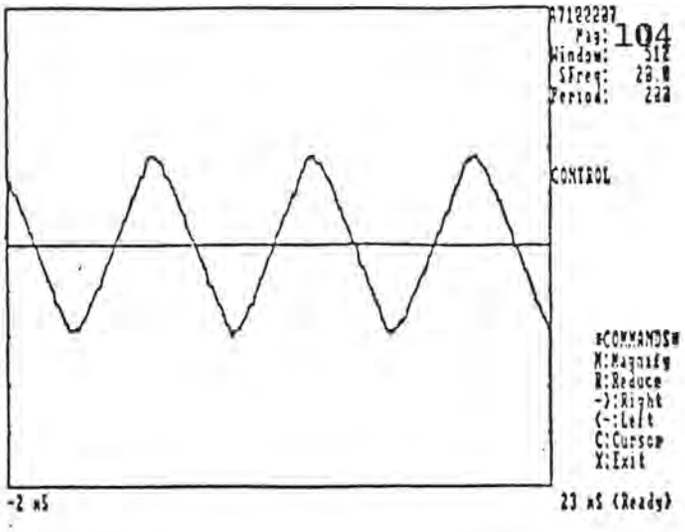


Figure 3.16F: Triangle waveform recovered by inverse filtering.

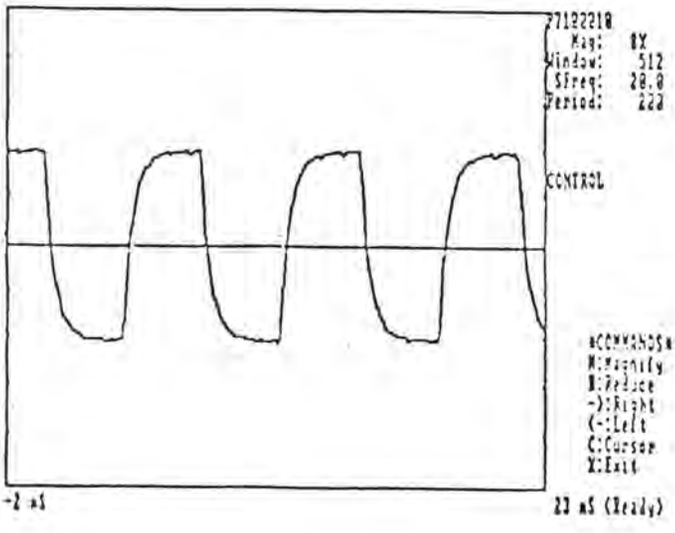


Figure 3.16G: Square waveform of 100 Hz, filtered by a high filter with f_C=340 Hz.

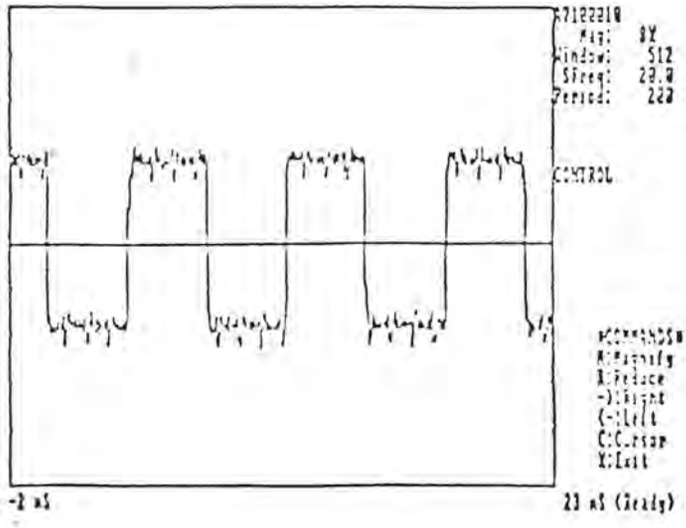


Figure 3.16H: Square waveform recovered by inverse filtering.

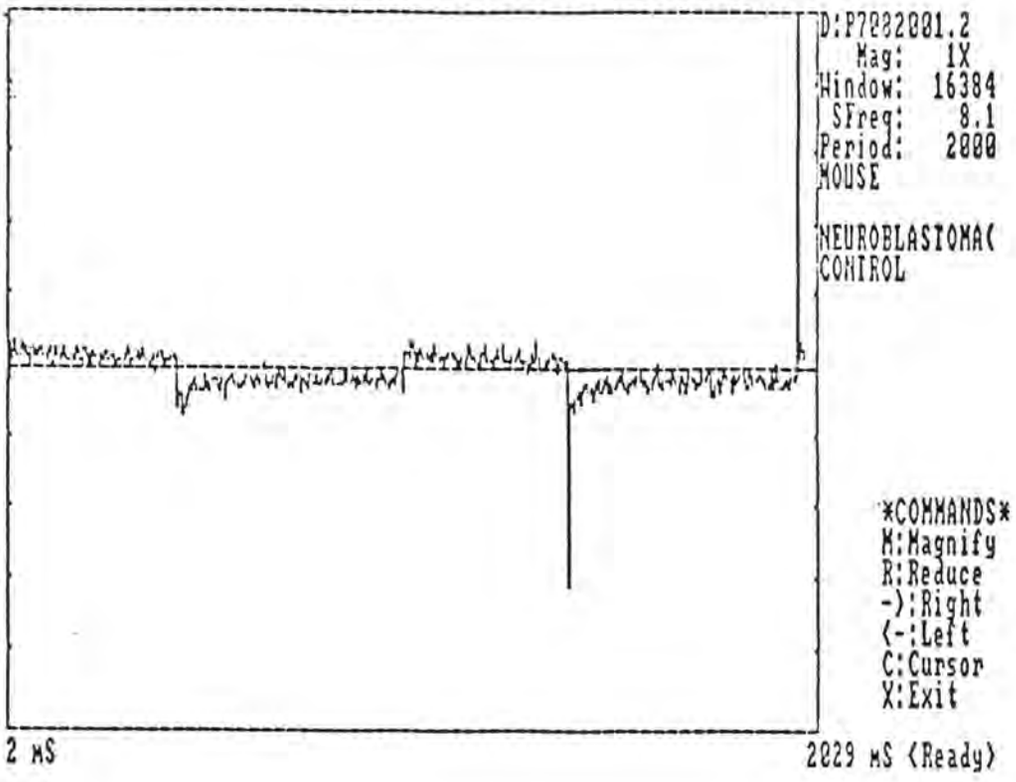


Figure 3.17A: A whole-cell recording made from a single neuroblastoma (N2A) cell, showing distinctive capacitive discharge spikes.

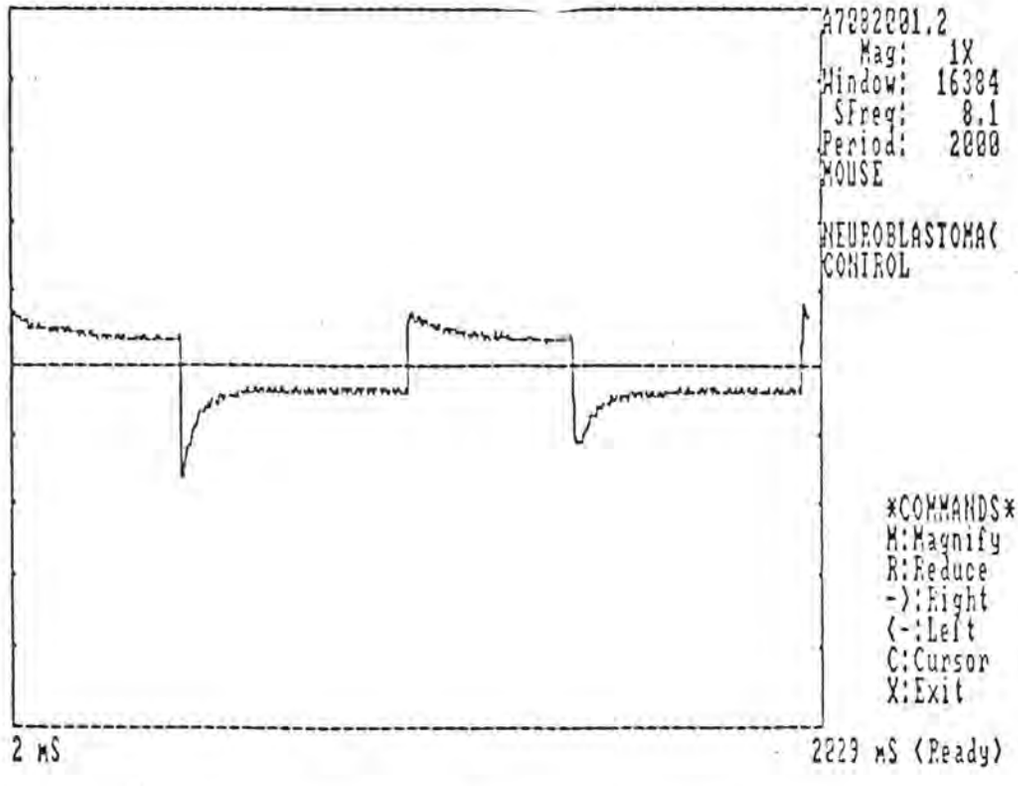


Figure 3.17B: The recording of Figure 3.17A, after inverse filtering. The sharp reduction in high frequency noise is the result of enhancement of lower frequencies by the inverse transform.

White noise analysis provides information similar to the frequency response method and an interesting comparison of the two techniques run on identical test setups is depicted in Figures 3.18a-c.

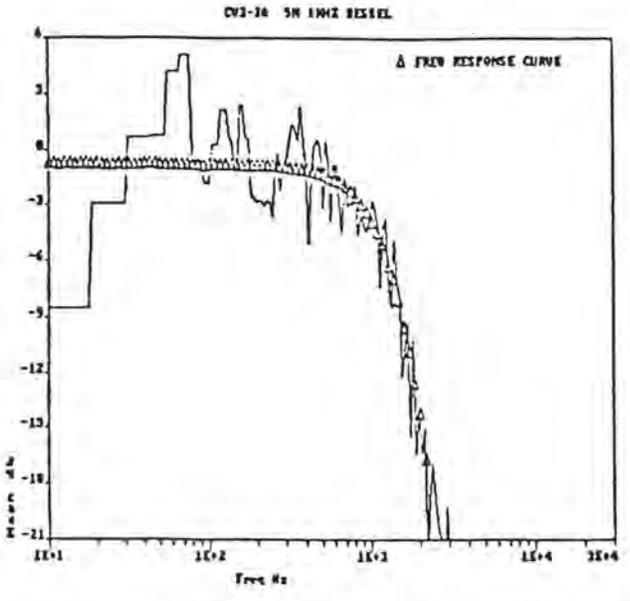


Figure 3.18A: Low pass filter, f_C=1 kHz.

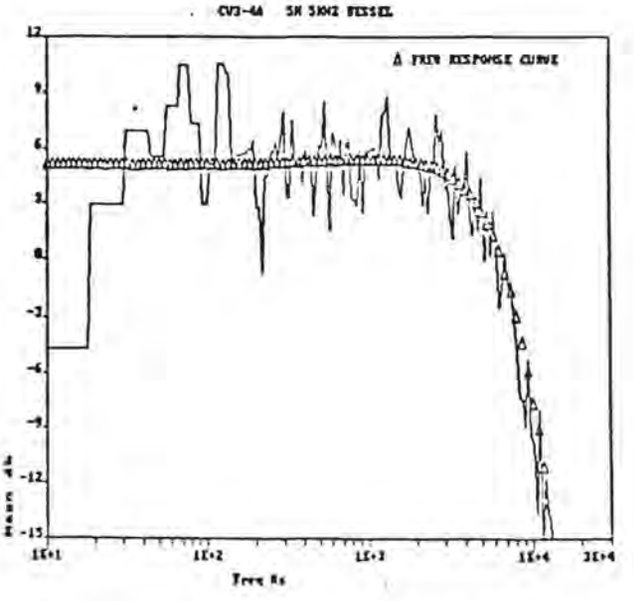


Figure 3.18B: Low pass filter, f_C=5 kHz.

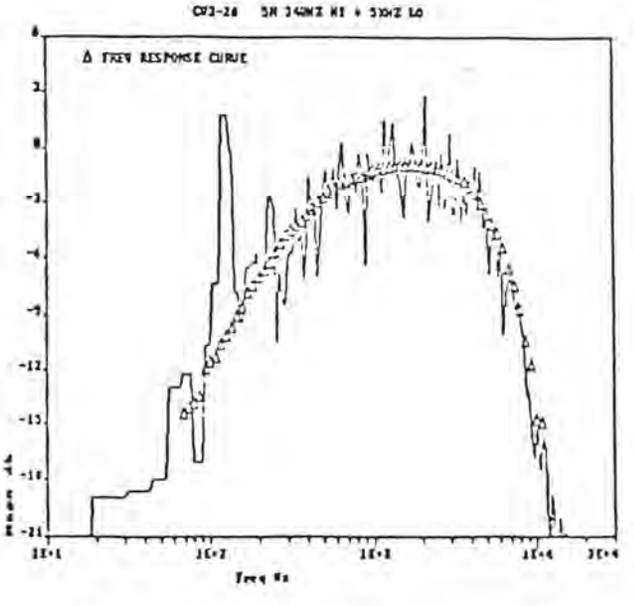


Figure 3.18C: Band pass filter, $f_{c,lo}$=340Hz, $f_{c,hi}$=5kHz. These spectra demonstrate that the results of white noise and frequency response measurement are in close agreement.

Each of the white noise curves was generated by taking eight data files of 4000 points sampled at 50 kHz, multiplying each file with a Hamming window and averaging the Fourier transform magnitudes. The fluctuations in the curves can be reduced still further by averaging a larger number of data records, but at 1.2 minutes per FFT the computation time requirement quickly becomes prohibitive.

The major drawback we have found in using the frequency response approach for system identification is that it takes time, occasionally as long as ten minutes, to make the large number of measurements needed to ensure good resolution. At present, we make 30 measurements in each of four frequency decades ranging from 10 Hz to 100 kHz for a total of 120 data points. When the system is noisy it takes longer for the detectors to stabilize sufficiently to obtain a reliable data point. Random noise added to the sine wave generally does not affect the validity of the analysis since the noise tends simply to offset the frequency response curve up by a constant magnitude, and the characteristics of the system are found not from the actual magnitude of the curve but from measurements of changes in the slope of the curve.

To carry out the inverse transform functions, the first recording from each set of recordings is processed with several different values for the artificial cutoffs p_1 and p_2. If the cutoffs are extended too far, overcompensation shows up as reverse filtering. In other words, if the high pass filter effect is overcompensated the resulting signal appears to have been passed through a low pass filter. Once a good set of artificial cutoffs is found, those values usually can be applied

without further modification to a series of recordings made under the same conditions. In some cases where especially fine resolution of data points is required, it may be necessary to determine optimal p_1 and p_2 values for each recording. There appears to be a practical limit on the amount of compensation which can be applied in a given situation and the ability to repeat the process as necessary to obtain optimal results is one of the strengths of the digital approach.

It should be noted that the high pass filter corner frequency determined during system identification frequently falls above the Nyquist frequency of the desired sampling rate. For example, a typical experimental arrangement may produce a high pass corner at 30kHz, while the experimenter wishes to set the sampling rate at 20kHz. In this situation the cutoff frequency parameter for inverse filtering should be set at the Nyquist frequency of 10kHz rather than the actual corner frequency. Use of an inverse filter cutoff frequency greater than half the sample rate produces unpredictable, or no discernable, results.

Review of the data records in Figure 3.16a-h reveals some interesting points. The corrected records in Figures 3.16b and 3.16d show slight magnitude distortion in the first few milliseconds of the record, after which this effect disappears entirely. This magnitude error is attributed to the fact that since the transfer function being applied to the record has a finite time constant, the full effects of the convolution are not manifested until essentially all of the transfer function can be employed in the convolution. Even though the inverse filtering is applied via the finite difference equations to enhance computational speed, the effect is the same as if the actual

convolution summation was implemented. The time taken to overcome this "end effect" is therefore directly related to the time constant of the inverse transfer function and to the magnitude of correction being undertaken.

Distortion of the square wave shown in Figure 3.16c is quite severe since the high pass cutoff frequency, 340Hz, is well above the square wave cycle frequency of 100Hz. To recover the square wave of Figure 3.16d, the distorted record was first overcompensated by inverse high pass filtering with inverse filter cutoff frequency of 340Hz and p_2 set to 10Hz. This corrected record was then processed with inverse low pass filtering, p_1=100Hz and p_2=7200Hz, to correct for the overcompensation. This pair of records, representing a case of extreme distortion of the desired signal, demonstrated the power and flexibility of the digital compensation technique.

The pair of records in Figure 3.16e-f are almost identical, since filtering a 100Hz triangle wave with a low pass filter of f_c=340Hz creates very little distortion. Figure 3.16h is a good demonstration of the ability of this method to recover lost information. When the record in Figure 3.16g was taken, the electrode holder was picking up electromagnetic interference with a periodicity of about 1ms, apparently the same 1 kHz interference mentioned earlier. This pattern was noted on the monitor oscilloscope at the input to the filter but was obscured by the filter. Gratifyingly, inverse filtering recovered not only the square wave but also the high frequency signal riding on it which, although unwanted, still serves to highlight the potential of digital signal enhancement.

When the analog compensation controls are properly set the experimenter usually sees, in real time, a good representation of the signal of interest. In this respect the analog methods are superior to the digital approach since with current technology microelectrode recordings must be stored and then processed ex post facto. This leads us to the next logical step in integrating digital signal processing methods with traditional electrophysiology. It was noted earlier that the settings on the analog controls could provide the parameters needed for inverse transform filtering. Conversely, the frequency response curve can tell the operator approximately where to set the analog controls for best results. It would therefore seem reasonable first to measure the frequency response of an experimental setup, then use that information to assist in setting the analog compensation controls, and then take another frequency response curve to determine how well the system is set up. After recordings have been digitized and stored, inverse filtering can be used to further enhance and refine the data. This "hybrid" combination of analog and digital methods may provide the best of both worlds. In experiments involving transmembrane impalements of whole cells to measure action potentials, digital enhancement of recordings probably will be of little benefit since in most cases only a small amount of negative capacitance compensation is needed to produce signals which are entirely adequate. However, the results may be quite dramatic when dealing with whole cell and patch clamp measurements of such phenomena as single fast sodium channel activity.

SUMMARY

Step function, frequency response and white noise methods of system identification have been applied to excitable cell membrane electrophysiology experiments. It appears that the frequency response approach can provide reliable estimates of the parameters of cell electrical models such as Figure 3.1b. Digital techniques for removing distortion from digitized recordings have been tested and successfully implemented. Discrete numerical analysis techniques such as that employed by Palti and Cohen-Armon (1982) have demonstrated the potential of the laboratory computer for data manipulation, and since microcomputers are increasingly available for use in the laboratory it is certain that digital signal processing methods will be in general use in the not too distant future. The ability to play back and post process recorded information as many times as desired is a crucial advantage of digital versus analog methods.

CHAPTER IV: MODELING AND SIMULATION OF MEMBRANE SINGLE ION CHANNEL ACTIVITY

INTRODUCTION

The channels or pores through which ions cross the cellular covalent lipid bilayer membrane are large, monolithic molecular units, imbedded in the bilayer, which may have two or more subunits. Channel activity in terms of "opening" to permit the flow of ions and "closing" to bar the passage of ions through the membrane is believed to correspond to changes in the physical conformation of the channel molecule. Factors which affect the types of transitions between conformational states and the rates at which they occur include temperature, transmembrane electrical potential and rate of change thereof, the presence or absence of other ion species, and various biochemical and pharmacological substances which inhibit or facilitate state transitions (Catterall et al 1986; Hille 1984; Läuger 1983; Moczydlowski et al 1986). Since the ion channels are directly responsible for cell electrical excitability, and therefore ultimately for both normal and abnormal electrophysiology (Hille 1984), channel activity and the associated controlling factors are the subjects of intense research. Models of ion channel behavior assist in understanding the physical, chemical and electrophysiological nature of the channels.

MODELS

In the simplest case the channel has only two physical conformations which correspond to the open and closed states, and is capable of reversible transitions between conformations. The reaction mechanism between open and shut states is described by a Michaelis-Menten relation:

$$\text{open} \underset{B}{\overset{A}{<=>}} \text{shut}$$

where A and B are the rate constants for the state transitions. Given this situation the time course of channel activity is described by determining the values of the rate constants as functions of the controlling factors described above. Where the rate constants depend mainly on transmembrane potential, as in the sodium channel, the values are determined by holding the transmembrane potential constant, recording single channel activity and plotting histograms of open and closed dwell times to estimate the probability density functions of the state transitions.

The form of the probability density functions can be predicted by making the following assumptions:

(1) Dwell times are random, independent, identically distributed variables, and there is no interaction between neighboring channels.

(2) Channel characteristics are "wide sense stationary" in that the distributions giving rise to the dwell times are

completely equivalent in at least the first two moments.

(3) The probability that two or more state transitions occur on an arbitrarily short time interval is essentially zero. That is,

$$\sum_{x=2}^{\infty} p(x,\Delta t) = o(\Delta t) \tag{1}$$

where $o(\Delta t)/\Delta t$ is any function that tends to zero as Δt approaches zero.

(4) The probability that a transition will occur on a given time interval is proportional to the length of the interval.

This model represents a "completely random" case in that the channel dwell times are assumed to be independent random variables. If $p(x,\Delta t)$ represents the probability that x events (state changes, i.e. transitions between open and closed states) occur on an interval Δt, then from the fourth assumption, above, the probability that one event occurs on Δt is expressed as

$$p(1,\Delta t) = a\Delta t + o(\Delta t) \qquad a > 0, \quad \Delta t > 0. \tag{2}$$

The probability that at least one event occurs on Δt is the sum of the probability that one event occurs plus the probability that more than one event occurs:

$$p(1,\Delta t) + \sum_{x=2}^{\infty} p(x,\Delta t) = a\Delta t + o(\Delta t) + o(\Delta t)$$

$$= a\Delta t + o(\Delta t). \tag{3}$$

and the probability that no events occur can be written as

$$p(0,\Delta t) = 1 - a\Delta t - o(\Delta t). \tag{4}$$

We can now discuss the probability that zero events take place on the

sum of two intervals, $t + \Delta t$, as the product of the probability that no events occur on each interval. That is,

$$p(0,t+\Delta t) = p(0,t)p(0,\Delta t)$$

$$= p(0,t)[1 - a\Delta t - o(\Delta t)]. \quad (5)$$

Expanding the right hand side of (5) yields

$$p(0,t+\Delta t) = p(0,t) - ap(0,t)\Delta t - p(0,t)o(\Delta t). \quad (6)$$

Subtract $p(0,t)$ from both sides,

$$p(0,t+\Delta t) - p(0,t) = -ap(0,t)\Delta t - p(0,t)o(\Delta t) \quad (7)$$

and divide both sides by Δt to give

$$\frac{p(0,t+\Delta t) - p(0,t)}{\Delta t} = -ap(0,t) - \frac{p(0,t)o(\Delta t)}{\Delta t}. \quad (8)$$

Now taking the limit of both sides as Δt approaches zero results in the differential equation

$$p'(0,t) = -ap(0,t). \quad (9)$$

The solution to this differential equation is

$$p(0,t) = ce^{-at} \quad (10)$$

and the condition that $p(0,0)=1$ implies that $c=1$ so that we have

$$p(0,t) = e^{-at}. \quad (11)$$

Extending this derivation to $x>0$, it is seen that

$$p(x,t+\Delta t) = [p(x,t)][1-a\Delta t-o(\Delta t)]+[p(x-1,t)][a\Delta t+o(\Delta t)]+o(\Delta t) \quad (12)$$

As in (7) and (8), subtract $p(x,t)$ from both sides of the equation, then divide both sides by Δt to yield

$$\frac{p(x,t+\Delta t) - p(x,t)}{\Delta t} = -ap(x,t) + ap(x-1,t) + \frac{o(\Delta t)}{\Delta t}. \quad (13)$$

Again taking the limit as Δt tends toward zero produces a differential equation

$$p'(x,t) = -ap(x,t) + ap(x-1,t) \quad (14)$$

for all positive integer x. By setting as boundary conditions that the p(x,0)=0 for all x, this system of differential equations can be solved. The result,

$$p(x,t) = (at)^x e^{-at} / x! \quad (15)$$

is in the form of a Poisson probability density function, and the probability that exactly one event occurs during the time interval t is

$$p(1,t) = ate^{-at}. \quad (16)$$

This analysis is supported by a conceptually different approach which gives the same result. Consider the channel molecule to be in an equilibrium condition in which the energy required to change states is supplied by random thermal energy. That is, each of the two states is a kinetic energy minimum, and there is an energy barrier between states. The molecule is in motion through numerous degrees of freedom, constantly bending, stretching, twisting and rotating, driven by random thermal energy. At some point sufficient thermal energy is present to overcome the barrier between states and a state transition occurs. Each motion can be viewed as a Bernoulli trial in which a successful outcome is a transition. Since molecular movement occurs on a picosecond scale while state transitions occur in terms of milliseconds there will be a very large number of trials, each with a very small probability of success, between transitions. The limiting case for a Bernoulli process with large n and small p is the Poisson distribution, ate^{-at}, as above.

The completely random model works well for the simplest case as described (Colquhoun, Hawkes 1983) but becomes increasingly complex, unwieldy and difficult to analyze from experimental data as the number

of possible states increases. Consider an ion channel with two closed states, in one of which the channel is shut because it is blocked by the binding of a ligand molecule. The rate reaction mechanism is defined as

$$\text{shut} \underset{A}{\overset{B}{<=>}} \text{open} \underset{k_{-B}}{\overset{k_{+B}}{<=>}} \text{inactivated} \qquad (17)$$

The histogram of the open dwell times is exponentially distributed with mean $1/(A+k_{+B}X_B)$, where X_B is the concentration of the blocker, but the distribution of closed times is now a mixture of exponentials representing the two closed states. Sorting out the coefficients of combined exponentials from a single experimental histogram is a difficult problem which does not have a rigorous solution. The values of experimentally determined coefficients are heavily influenced by variability in the data and in complex models the experimenter must have a good idea a priori of how many conformational states are present.

Consideration of the channel as a molecule with two minimum energy states separated by an energy barrier facilitates a completely different analytical approach. The energy minima represent two attracting equilibrium points separated by a repelling point, the barrier, resulting in a system which exhibits "bistability". The series of open and closed dwell times can be described as a bistable time series generated by a stochastic catastrophe model (Cobb, Zacks 1987) of the form

$$x(t+1) = [2x(t) - wx(t)^3 + sU(t)]_{-A}^{+A} \qquad (18)$$

with $A = (1/w)^{\frac{1}{2}}$, $w>0$, and $s>0$. This is a nonlinear "censored" Markov process based on the cusp catastrophe model, with the results constrained within the domain of attraction [+A,-A]. The random terms U(t) are distributed normally with zero mean and variance of σ^2, representing random thermal energy perturbations of the system. The values of x(t) represent the location of the channel molecule on a potential energy curve at a given time t. A state transition occurs when a sufficient combination of random events pushes the process across the repelling point and into the vicinity of the other attractor. Note that +A and -A are repelling points, and if the system is pushed outside of these limiting values by the random perturbations it is unlikely to return to the domain of attraction, hence the requirement for constraints.

Figure 4.1 is a plot of the equation $x(t+1) = 2x(t) - x(t)^3$ which illustrates the concept of attracting and repelling points. Attractors correspond to the local maximum and minimum while the repeller, in this case located at the origin, is the barrier between them. Open and closed dwell times are the intervals spent by the molecule in the vicinity of the attraction equilibra. Since the local maximum and minimum are equal in magnitude, there is no preference of one state over the other and the amount of time spent in each state is equal.

This is an attractive concept. The model is easily extended to multiple states by increasing the order of the equation to add local maxima and minima corresponding to additional attractors and repellers. Fitting the model to experimental data will provide direct information on the relative energy levels of various molecular conformations. This

in turn will facilitate selection between different models of molecular structure, between types of channels (e.g. fast and slow calcium current channels), and also should assist in determining how many and what types of conformational states are present. According to the preceding descriptions and for the simplest case, the completely random and nonlinear catastrophe models should be completely equivalent. If this is true, then simulated data sets generated using either model should be indistinguishable.

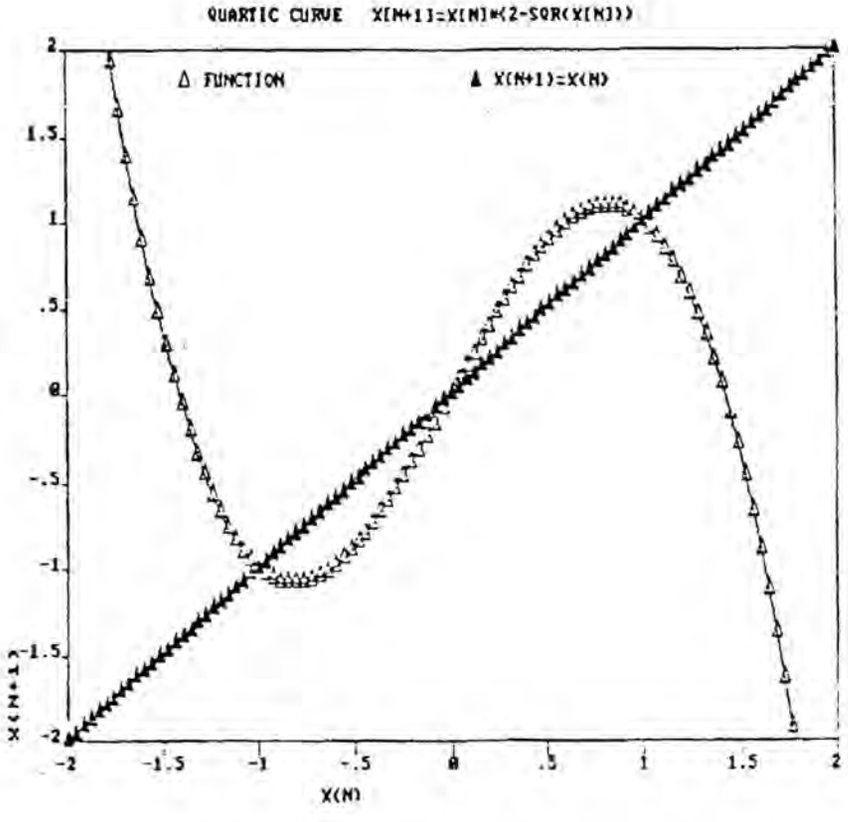


Figure 4.1: Graph of a quartic equation representing a bistable process. The points at which the local maximum and minimum cross the linear plot correspond to the attractors, and the zero crossing is the repeller.

A fundamental assumption of both models is that the state dwell

times are independent. Jackson et al (1982) have speculated that some channel events may be correlated in time, and the existence of such phenomena as flickering and bursts of openings also indicate possible correlations between successive events. A system in which the present value of x(t) depends to some extent on the previous value, x(t-1), is often referred to as a first order Markov process. An equation describing this process is

$$x(t) = ax(t-1) + e(t) \qquad (t>0) \qquad (19)$$

where e(t) is a random variable with zero mean and variance σ^2. The coefficient a describes the strength of the relationship between the present event and the event immediately preceding. This equation does not apply exactly to this case because all of the x(t) must be greater than zero (obviously there cannot be a negative "time" spent in a particular state) and the presence of the e(t) terms gives rise to a nonzero probability that the right hand side of the equation will not be positive. There are two modifications which can be made to ensure that the x(t) are all positive. One is to ensure that the probability of a nonpositive result is small and then constrain the process to positive x(t). The equation would then take the form

$$x(t) = [ax(t-1) + e(t)]_{d}^{+\infty} \qquad (t>0,\ d>0) \qquad (20)$$

with the e(t) distributed randomly with mean greater than zero, and variance restricted to ensure that the probability of the random term exceeding the linear term is small. The constraint terms define the range of the process as $(d<x(t)<\infty)$, so that a result less than d is arbitrarily set equal to d.

The second modification involves transforming the data to ensure positive x(t) values. This can be accomplished by defining $y(t)=x(t)^2$ so that

$$y(t) = a^2x(t-1)^2 + 2ax(t-1)e(t) + e(t)^2 \qquad (21)$$

In this equation the present dwell time y(t) depends on a linear combination of transformations of previous dwell times and random terms. The random elements e(t) may now be distributed normally with zero mean and variance of σ^2. It is difficult to relate the terms in the right hand side of the second modification (21) to properties of the single channel, and this expression is therefore purely a descriptive device. For this reason the first modification (20) is the better model despite the artificiality of the constraint terms. Proper choice of the coefficients and random distribution properties will minimize the effect of the constraint on the results.

TESTS

Several methods of testing the assumption of the independence of consecutive dwell times x(t-1) and x(t) are available. One technique consists of setting x(t-1) and x(t) equal to x(i) and y(i) respectively, then computing from linear regression the coefficients of the equation

$$y(i) = a + bx(i)$$

A test is conducted to determine whether b=0. If so, then there is no linear relationship between x(i) and y(i) and the dwell times are independent within the confidence limits of the test. A problem with

this technique is that linear regression analysis assumes that the variables x(i) are not correlated (Neter et al 1983), while in fact the strength of correlation is exactly what we are testing. This difficulty can be minimized by realizing that if two successive variables are related by a coefficient a, (-1<a<1), then terms separated by a delay d are related by a^d, which rapidly diminishes with increasing d. This suggests that the problem of correlation between x(i-1) and x(i) can be minimized by using infrequent pairs of dwell times separated by a delay d. In using this approach much of the data will be ignored and, further, sufficient data points must be available to carry out the analysis.

A second technique derived from time series analysis (Priestly 1981) is to estimate the autocovariance sequence as

$$R(r) = (1/N) \sum_{t=1}^{N-|r|} x(t)x(t+|r|); \qquad 0 < r < (N-1) \qquad (22)$$

The autocorrelation sequence is defined as R(r)/R(0), and the first term, R(1)/R(0), is the correlation between x(t-1) and x(t). This estimate does not depend on an assumption regarding independence between variables, and in fact is intended specifically to determine the relationship not only between adjacent events but between any one event and a second event separated from the first by a delay of r. The better method for estimating the correlation coefficient can be decided on by generating a number of simulated data sets with various correlation coefficients, applying both tests to each set of data points, then computing the coefficient of determination, r^2, for the estimated and actual coefficients for each test. The better method will

produce a higher coefficient of determination. A further test can be applied to the coefficient of correlation, $r=[r^2]^{\frac{1}{2}}$, to determine whether the better estimation method is in fact an adequate estimator.

OBJECTIVES

This portion of the research effort has the following objectives:

(1) Develop algorithms for simulating dwell time data from each of the three models.

(2) Determine which of two methods, linear regression or autocorrelation estimation, is better for testing the hypothesis of independence of channel state dwell times.

(3) Use histograms and the method of (2) to determine whether the completely random and nonlinear catastrophe models produce indistinguishable results, as hypothesized.

METHODS

Channel state dwell times produced by the simulation algorithms must result in histograms which show an exponential distribution, with mean and standard deviation both equal to 1/a, to be in consonance with experimental data. With sodium channel open state dwell times as an example, a range of 2 to 6 ms is used for these simulations (Fenwick, Marty and Neher 1982; Jackson, Lecar 1979; Neher, Sakmann 1976; Sigworth, Neher 1980).

SIMULATIONS

Common to all three models is the random variable term. All programs were written in Turbo PASCAL (Borland) which has a pseudorandom number generator. This function is designed to produce random numbers which are uniformly distributed between 0 and 1. These numbers are converted to exponentially distributed variables by taking the natural logarithm of each number. These in turn can be distributed evenly around zero by again calling the random function, then reversing the sign of the transformed number if the second number happens to be less than 0.5.

In the case of the completely random model the second step is not necessary. The random variables corresponding to dwell times are generated as

$$x(i) = -m*\ln(\text{random}); \qquad 2 < m < 6 \qquad (23)$$

where m=1/a, in the range given, produces data with mean and standard deviation between 2 and 6.

The nonlinear catastrophe model is simulated from the equation

$$x(i+1) = x(i)(2-(wx(i)^2))+(dr); \quad w = 1; \quad 0.115 < d < 0.250 \qquad (24)$$

where r is a random variable generated as described above. This produces a bistable function with attractors at ±1.414. The function is constrained to within ±1.63 to keep the values of x(i) within the domain of attraction. The value of 1.63 is 1.5 times the limit of the range of attraction, and was chosen so as to allow the function to stray beyond the range of attraction, but not too far. This function represents a channel in the open state while the values of x(i) are greater than zero, and a time interval counter is incremented by 0.1 ms

for each consecutive iteration for which x(i) is positive. To preserve the continuity of the function, iterations are continued while x(i) is negative, and the counter is restarted for a new dwell time when x(i) again becomes positive. Figures 4.2a and 4.2b are segments from the nonlinear model drawn while the model was generating simulated data sets with mean dwell times of 2 ms and 6 ms, respectively. These plots bear striking resemblance to single channel recordings (Colquhoun, Hawkes 1983).

The first order linear or Markovian model is implemented as

$$x(i) = L+ax(i-1)+(dr); \quad 0.1 < a < 0.5; \quad 0.8 < d < 1.2 \qquad (25)$$

with r a random variable as above.

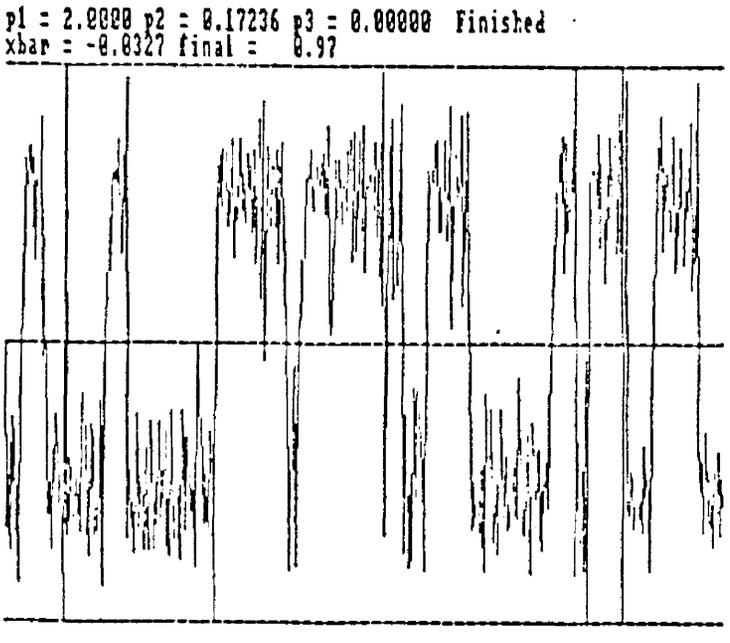


Figure 4.2A: Time series plot generated by the model given in (24). Mean dwell time of this simulated data set is 2.0 ms. Upper and lower bounds in 4.2A and 4.2B mark the censoring limits of the process.

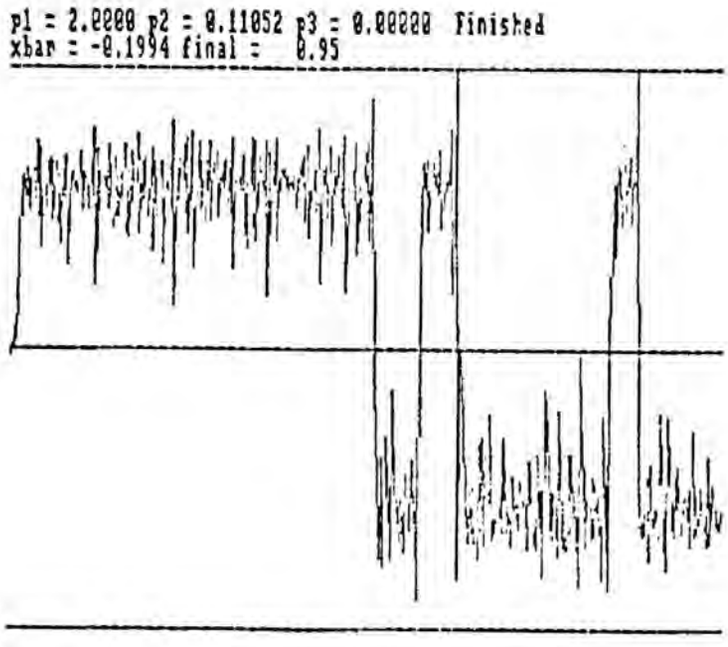


Figure 4.2B: Time series plot from (24) with average dwell time of 6.0 ms.

ANALYSIS

Autocorrelation estimates for the first fifty terms (0 r 50) of each data set are computed using the summation sequence described earlier. Since the completely random and nonlinear catastrophe models are expected to show no significant correlations beyond the R(0) term, and the first order Markov model is limited to a maximum first order correlation of 0.5, fifty terms in the autocorrelation sequence are more than adequate to specify the behaviour of the data sets.

The regression coefficient b is calculated as (Neter et al 1983)

$$b = \Sigma [(X(i)-\bar{X})(Y(i)-\bar{Y})]/\Sigma (X(i)-\bar{X})^2 \qquad (26)$$

The coefficient of determination between estimated and actual values of

the correlation coefficient in the Markov model is calculated as

$$r = [\Sigma(X(i)-\bar{X})(Y(i)-\bar{Y})]/[\Sigma(X(i)-\bar{X})^2\Sigma(Y(i)-\bar{Y})^2]^{\frac{1}{2}} \quad (27)$$

where the X(i) are the correlation coefficients used to produce the simulated data sets, and the Y(i) are the estimates generated by either the regression equation or the autocorrelation sequence. The significance of the coefficient of correlation, r, is tested by computing a t-test value from

$$t = r[(n-2)/(1-r^2)]^{\frac{1}{2}} \quad (28)$$

and comparing this value with a tabulated value using (n-2) degrees of freedom.

RESULTS

All of the models were tested to determine the range of parameters needed to generate simulated data sets with mean dwell times in the range of 2 to 6 ms. The simulation program generated 123 files from the random model, 90 from the nonlinear catastrophe model and 105 files from the first order Markov model. Each file consisted of 1000 channel dwell times. In the case of the Markov model, the correlation coefficients were programmed in groups in the range of 0.1 to 0.5, and 21 files were generated for each group. Linear regression was used to match the coefficients in the models to statistics calculated from the data sets with the following results:

RANDOM:

$$x(t) = De(t)$$

$$\bar{x} = -0.0216 + 1.0062D \; ; \quad R^2 = 0.9908$$

$$\bar{s}_x = -0.0474 + 1.0092D \; ; \quad R^2 = 0.9833$$

NONLINEAR:

$$x(i+1) = 2x(i) - x(i)^3 + De(i)$$

$$\bar{x} = 0.3679D^{-1.7436} \; ; \quad R^2 = 0.9741$$

$$\bar{s}_x = 0.2907D^{-2.0301} \; ; \; R^2 = 0.9827$$

MARKOV:

$$x(t+1) = L + Ax(t) + De(t)$$

where $D = L / (1 - A)$

$$\bar{x} = -0.5083 + 1.6024L + 1.7190D \; ; \quad R^2 = 0.9905$$

$$\bar{s}_x = -0.3154 + 1.0608L + 1.1310D \; : \; R^2 = 0.9775$$

Another requirement for the simulations to be valid, since the process being simulated has a logarithmic probability density function, is that there must be a 1:1 correspondence between mean and standard deviation. First order linear regression between mean and standard deviation yielded values of 1.001 for the random model, 1.089 for the nonlinear catastrophe model and 0.71 in the Markov case. A low value for the Markov model is not unexpected since the magnitude of each dwell time is by definition dependent to some extend on previous values, reducing the randomness and hence the variability of the data. The value of 1.089 for the nonlinear model is slightly high, and is dependent to some extent on the choice of 2 as the first order coefficient in the model (24). Experiments indicate that the model is very sensitive to changes in this coefficient since the magnitude of the local maximum and minimum depend directly upon it. The value of 2 was chosen because the resulting maxima and minima are well defined, and therefore so was the bistable behavior of the model, and because the resulting ratio of

mean to standard deviation was very close to unity.

The 105 files generated from the Markov model were used to compare the effectiveness of the linear regression and autocorrelation sequence methods for estimating the correlation coefficient. Values for the correlation coefficients were estimated from each file by each method, and coefficients of determination (R^2) used to measure the strength of the relationship between actual and estimated coefficients. The first term of the autocorrelation sequence, R(1)/R(0), gave an R^2 value of 0.5318, while the R^2 from regression was 0.3287. However, since the regression used only every tenth pair of values to minimize correlations between variables assumed to be independent, only 10% of the data were utilized. In computing the autocorrelation sequence, 100% of the data are available for use in estimating R^2. If the problem of correlations between the independent variable data points is ignored the regression calculations are performed using all pairs of data points, the computation yields an R^2 of 0.5269. This is essentially the same as that computed from the autocorrelation sequence.

DISCUSSION

All simulated data files were characterized by histograms of the channel dwell times and graphs of the first 25 terms of the autocorrelation sequences. As can be seen in the dwell time histograms in Figure 4.3, the random and nonlinear models are very similar while the Markov model curves are visually quite distinct. The Markov data sets are increasingly dissimilar from the other two in proportion to

the magnitude of the correlation coefficient.

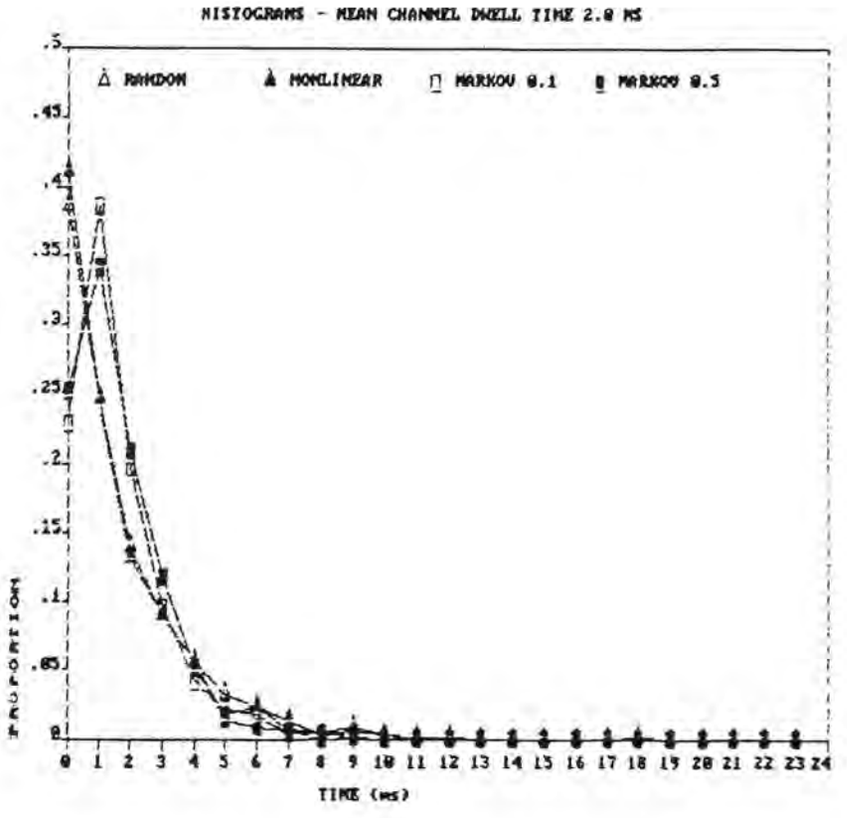


Figure 4.3A: Dwell time histograms from simulated data sets with average dwell times of 2.0 ms. Two histograms are from the Markov model using correlation coefficients of 0.1 and 0.5.

This is not a surprising result considering the nature of the models. The appearance of dwell time histograms with the morphology of the Markov data sets may be taken as an indication that the dwell times are correlated. Figure 4.4 shows the autocorrelation sequences for the same data as Figure 4.3. The curves for the Markov data, markedly so for correlation coefficient of 0.5, clearly show nonzero first terms as opposed to those for the random and nonlinear models, demonstrating that the autocorrelation sequence can be used to test the validity of the assumption of independence between successive dwell times.

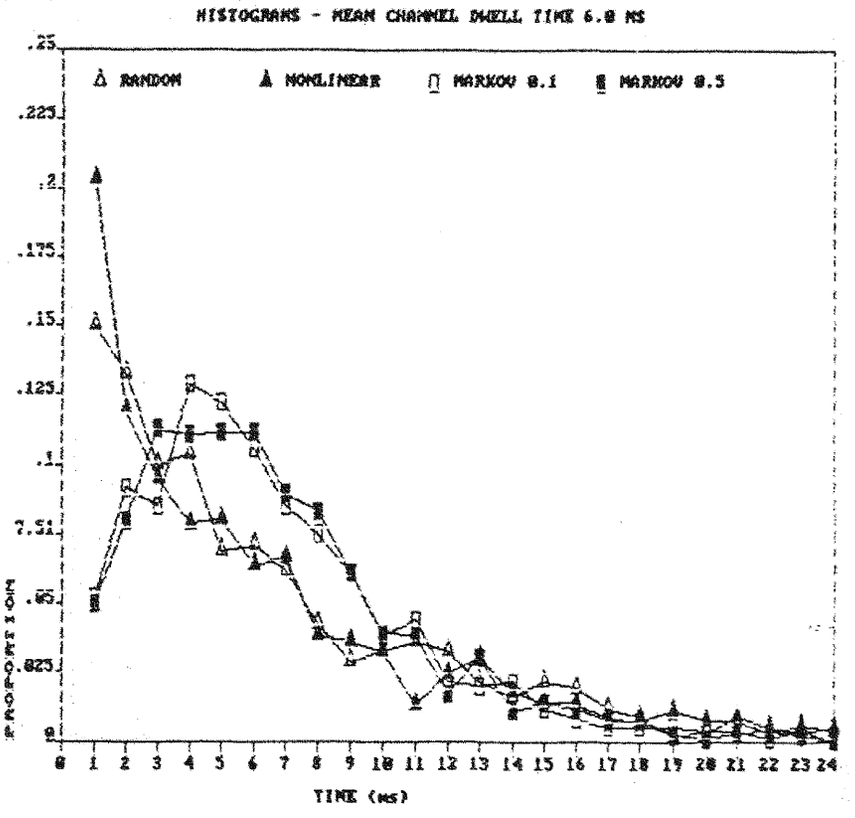


Figure 4.3B: Dwell time histograms from simulated data using each of the three models. Average dwell time in these data sets is 6.0 ms.

The fact that the autocorrelation sequence and regression analysis resulted in essentially the same R^2 values might lead to the conclusion that, for small correlation coefficients, the two methods are equally suitable. The autocorrelation sequence is the better method since not only does it not violate any assumptions, it provides information about correlations for the entire data set. This enhances the possibility of identifying different types of systems, such as autoregressive moving average (ARMA), or higher order Markov models which may be operative. Also, any periodicity in the data will appear in the autocorrelation

sequence. Finally, the relationships between successive events in different states can be explored using the closely related crosscorrelation sequence.

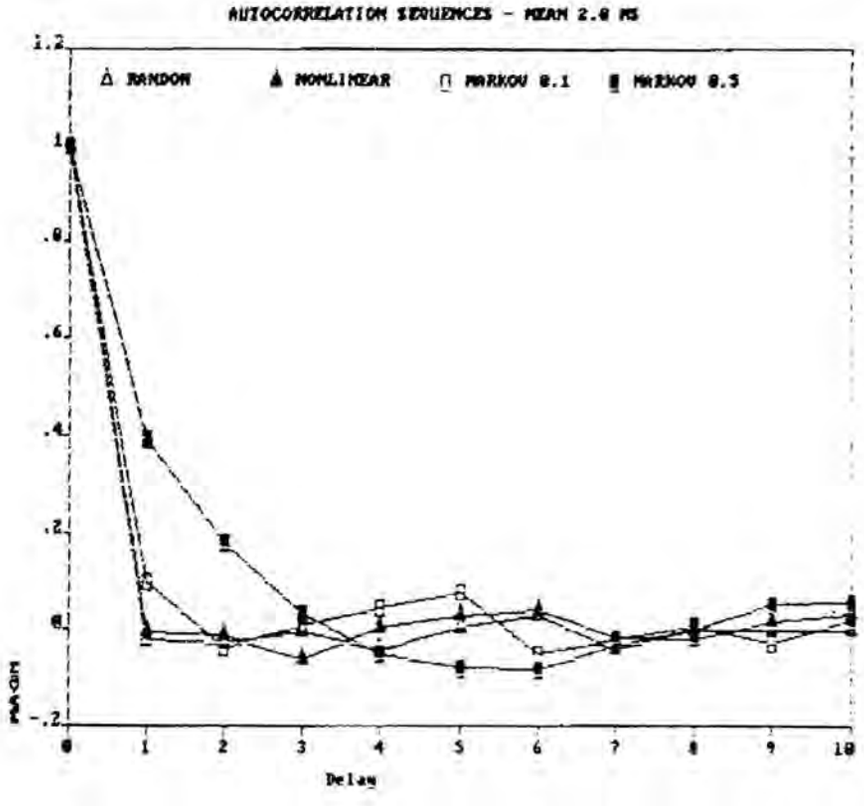


Figure 4.4A: The first ten elements of the autocorrelation sequences computed from the data sets depicted in Figure 4.3A.

The information presented in Figures 4.3 and 4.4 indicate that within the range of this type of presentation the random and nonlinear catastrophe models produce simulated data that are indistinguishable. This signifies that the nonlinear approach is as appropriate to the data as the traditional completely random model for this simplest case. As discussed earlier, the nonlinear model is easily extended to multiple open, closed and inactive states. Additional states are

incorporated into the model by raising the order of the polynomial one degree for each extra state. The relationship between model parameters and experimental data given in the results section provides a direct link between observed mean dwell times and the relative magnitude of the energy barrier between states.

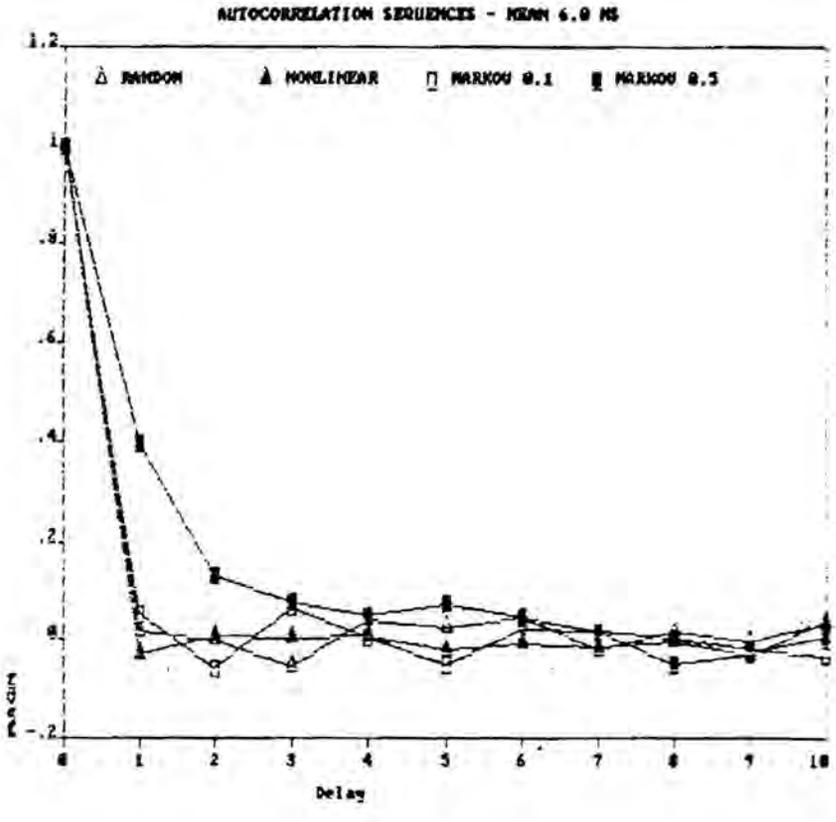


Figure 4.4B: The first ten elements of the autocorrelation sequences computed from the simulated data sets depicted in figure 4.3B.

Absolute values for the kinetic energy of various states in a multi-state system can be estimated if the kinetic energy of a single state is known. Adjusting the roots of the model equation can account for inequalities in kinetic energy levels of different states. Note that

the random model analyzes one state at a time, taking no account of the existence and characteristics of other states, while the nonlinear model encompasses all states in a unified descriptor. All of the above provide the basis for ongoing research in applying this model to present investigations into the nature of membrane ion channels.

SUMMARY

The autocorrelation sequence has been shown to be an effective indicator of the level of interdependence between successive channel events. This provides evidence that the crosscorrelation sequence may be useful for investigating similar relationships between sequential events from different states, i.e. channel closure followed by channel opening as examined by Jackson et al (1982).

Simulated data indicate that the nonlinear model derived from catastrophe theory is a valid approach for investigating the nature of conformational states and state transitions. All possible states are included in the model, and parameter fitting provides information about the relative kinetic energy levels of different states and the energy barriers between them.

LIST OF REFERENCES

Bigger, J.T., A. C. Basset, and B. F. Hoffman. "Electrophysiological effects of diphenylhydantoin on canine Purkinje fibers." Circ. Res. 22; 221-236, 1968.

Binstock, L., W. J. Adelman Jr, J. P. Senft, and H. Lecar. "Determination of the resistance in series with the membrane of giant axon." J. Mem. Biol. 21; 25, 1975.

Brown, P. B., and R. J. Millecchia. "A simple microprocessor-based laboratory controller. I. Hardware and firmware." J. Electrophysiol. Techs. 10; 159-177, 1983.

Buckles, D., and K. W. Hewett. "Assessment of the maximum frequency components and digital sampling of cardiac Purkinje fiber action potentials." Comp. Bio. Research. 19; 410-416, 1986.

Burden, R. L., and J. D. Faires. Numerical Analysis. Boston: Prindle, Weber & Schmidt, 1985.

Catterall, W. A., J. W. Schmidt, D. J. Messner, and D. J. Feller. "Structure and Biosynthesis of Neuronal Sodium Channels". In: Tetrodotoxin, Saxitoxin and the Molecular Biology of the Sodium Channel. New York: The New York Academy of Sciences, 1986, pp. 186-191.

Conte, S. D., and C. de Boor. Elementary Numerical Analysis. New York: McGraw-Hill Company, 1980, pp. 297-299.

Colquhoun, D. and A. G. Hawkes. "The Principles of the Stochastic Interpretation of Ion-Channel Mechanisms." In: Single-Channel Recording, edited by B. Sakmann and E. Neher. New York: Plenum Press, 1983, pp. 135-145.

Diez Gil, J. L., J. R. Ramon, L. A. Lastra,and P. D. Garcia de Jalon. "An operative system for treatment of electropharmacological signals with a microprocessor." Comp. Progs. Biomed. 18; 247-258, 1984.

Elharrar, V., and D. A. Lovelace. "On-line analysis of intracellular electrophysiological data using a microcomputer system." Am. J. Physiol. 237; H400-H408, 1979.

Fenwick, E. M., A. Marty, and E. Neher. "Sodium and Calcium Channels in Bovine Chromaffin Cells." J. Physiol. 331; 599-635, 1982.

Finkel, A. AXOPATCH-1A PATCH CLAMP THEORY AND OPERATION. Burlingame, CA: Axon Instruments, Inc., 1986.

Jackson, M. B., and H. Lecar. "Single postsynaptic channel currents in tissue cultured muscle." Nature 282; 863-4, 1979.

Jackson, M. B., B. S. Wong, C. E. Morris, H. Lecar, and C. N. Christian. "Successive Openings of the Same Acetylcholine Receptor Channel Are Correlated in Open Time." Biophys. J. 42; 109-114, 1982.

Fusi F., G. Piazzesi, S. Amerini, A. Mugelli, and S. Livi. "A low-cost microcomputer system for automated analysis of intracellular cardiac action potentials." J. Pharm. Methods. 11; 61-66, 1984.

Hille, B. Ionic Channels of Excitable Membranes. Sunderland, MA: Sinauer Associates, Inc., 1984.

Hodgkin, A. L., A. F. Huxley, and B. Katz. "Measurement of current-voltage relations in the membrane of the giant axon of Loligo." J. Physiol. 116; 424-448, 1952.

Klein, D-L, J. M. Jenkins,and R. E. Ten Eick. "Dual microcomputer analysis of cardiac transmembrane action potentials." IEEE Trans. Biomed. Engr. BME-30 12; 819-825, 1983.

Läuger, P. "Conformational Transitions of Ionic Channels". In: Single-Channel Recording, edited by B. Sakmann and E. Neher. New York: Plenum Press, 1983, pp. 177-188.

Moczydlowski, E., A. Uehara, X. Guo, and J. Heiny. "Isochannels and Blocking Modes of Voltage-Dependent Sodium Channels". In: Tetrodotoxin, Saxitoxin and the Molecular Biology of the Sodium Channel. New York: The New York Academy of Sciences, 1986, pp. 269-291.

Moore, J. W., M. Hines, and E. M. Harris. "Compensation for resistance in series with excitable membranes." Biophys J. 46; 507-514, 1984.

Neher, E., and B. Sakmann. "Single-channel currents recorded from membrane of denervated frog muscle fibres." Nature, Lond. 260; 779-802, 1976.

Neher, E., and B. Sakmann. "Noise analysis of Drug Induced Voltage Clamp currents in Denervated Frog Muscle Fibres." J. Physiol. 258; 705-729, 1976.

Oppenheim, A. V., and R. W. Shafer. DIGITAL SIGNAL PROCESSING. New Jersey: Prentice-Hall Company, 1975.

Palti Y., and M. Cohen-Armon. "Numerical method for correcting the series resistance error in voltage clamp experiments." Isr. J. Med. Sci. 18; 19-24, 1982.

Sakmann, B., and E. Neher. SINGLE-CHANNEL RECORDING. New York: Plenum Press, 1983, pp. 345-488.

Sigworth, F. J. "The variance of sodium current fluctuations at the Node of Ranvier." J. Physiol. 307; 97-129, 1980.

Sigworth, F. J., and E. Neher. "Single Na^+ channel currents observed in cultured rat muscle cells." Nature 287; 447-9, 1980.

TABLE I - Comparison of Measurement Methods

(H - hand measurement, W - waveform analysis)

FILE #	MDP		AMP		V_{max}		APD_{50}		APD_{90}	
	H	W	H	W	H	W	H	W	H	W
001	88	87	126	128	668	623	150	152	217	224
002	86	86	124	122	832	882	137	140	184	192
003	92	92	125	125	550	518	182	181	244	243
004	87	87	128	128	617	618	163	159	206	207
005	90	90	127	127	679	667	160	158	209	208
006	88	88	125	127	770	714	162	158	213	208
009	96	96	130	131	647	637	177	173	251	245
010	96	97	127	129	625	686	177	178	247	254
011	74	74	112	112	447	492	188	189	244	249
013	89	90	129	131	630	680	177	175	250	251

TABLE II - Card Edge Pin Assignments

PIN	FUNCTION	RANGE
1	Oscilloscope trigger	0 to 5 VDC (TTL Level)
2	Sawtooth calibrator trigger	-15 to +15 VDC
3	Stimulus channel 1	0 to 5 VDC low impedance
4	Stimulus channel 2	0 to 5 VDC low impedance
5	Parallel data bit 0	TTL
6	Parallel data bit 1	TTL
7	Parallel data bit 2	TTL
8	Parallel data bit 3	TTL
9	Parallel data bit 4	TTL
10	Parallel data bit 5	TTL
11	Parallel data bit 6	TTL
12	Parallel data bit 7	TTL
13	Data transfer strobe bit	TTL
14	A/D acquisition trigger	0 to 5 VDC (TTL)
15	Not used	
16	MC6802 Reset line	TTL (Negated)
17	Clamping command voltage	-15 to +15 VDC
18	Non-maskable interrupt	TTL
19	+5 VDC supply	
20	Ground	
21	+15 VDC supply	
22	-15 VDC supply	

TABLE III - Parts List

QTY	DESCRIPTION	PART NO.
1	Bud Box	
1	+/- 15 VDC power supply	
1	+ 5 VDC power supply	SK9067
1	Heat sink for power supplies	
1	25DB chassis pin connector	
1	Circuit board	
1	44-pin card edge connector	
1	SPST power switch	
1	Power indicator light	
1	Fuse holder with 1/4A fuse	
6	Terminal connectors	
2	40 pin DIP sockets	
1	28 pin DIP socket	
1	24 pin DIP socket	
5	8 pin DIP sockets	
1	47K 1/8W resistor	
2	2.2K 1/8W resistors	
2	10K 1/8W resistors	
1	10uf 25 VDC electrolytic capacitor	
2	47pf disk capacitors	
1	D/A converter chip	ICL7134B
1	EPROM	TMS2516JL-45
1	Microprocessor chip	MC6802CP

1	Peripheral Interface Adaptor	SC44074PK/MC6820/MC6821
1	Quad NAND gate	SN7400
5	Operational amplifiers	LM741CN/TL071ACP
1	Power cord with plug	
	Miscellaneous mounting hardware and standoffs	

APPENDIX - PPU SOFTWARE LISTING

```
     1                *
     2                *
     3                * PERIPHERAL PROCESSOR SOFTWARE
     4                *
     5                * VERSION #: X5.01
     6                *
     7                * PROGRAMMER: D. S. BUCKLES
     8                *
     9                * REVISION DATE: 10 DEC 86
    10                *
    11                * SOFTWARE INTENDED FOR THE CARDIAC
                        ELECTROPHYSIOLOGY
    12                 * PERIPHERAL PROCESSOR BOARD. CONTAINS VOLTAGE
                        CLAMP
    13                * AND TRANSMEMBRANE ACTION POTENTIAL SUBSETS.
    14                * STORES IN 2516 EPROM.
    15                * DESIGNED FOR INTERFACE WITH DT2801A.
    16                *
1    **** PPU              ****
0
    17                        NAM             PPU
    18        0800            ORG     $0800
    19                *
    20                * INITIALIZE VARIABLE ADDRESSES
    21                *
    22                *
    23        0000    PRGSEL  EQU     $00
    24        0001    SCTRIG  EQU     $01
    25        0002    SWTRIG  EQU     $02
    26        0003    S1TRIG  EQU     $03
    27        0004    S2TRIG  EQU     $04
    28                *
    29        0005    SWDEL   EQU     $05
    30        0006    S1DEL   EQU     $06
    31        0006    P1DEL   EQU     $06
    32        0007    S1DURH  EQU     $07
    33        0007    P1DURH  EQU     $07
    34        0008    S1DURL  EQU     $08
    35        0008    P1DURL  EQU     $08
    36        0009    S1S2IH  EQU     $09
    37        0009    P1P2IH  EQU     $09
    38        000A    S1S2IL  EQU     $0A
    39        000A    P1P2IL  EQU     $0A
    40        000B    S2DURH  EQU     $0B
    41        000B    P2DURH  EQU     $0B
    42        000C    S2DURL  EQU     $0C
    43        000C    P2DURL  EQU     $0C
    44        000D    RHDURH  EQU     $0D
```

```
45          000E    RHDURL EQU    $0E
46          000F    RCYCH  EQU    $0F
47          0010    RCYCL  EQU    $10
48          0011    P1VH   EQU    $11
49          0012    P1VL   EQU    $12
50          0013    P12VH  EQU    $13
51          0014    P12VL  EQU    $14
52          0015    P2VH   EQU    $15
53          0016    P2VL   EQU    $16
54          0017    RHVH   EQU    $17
55          0018    RHVL   EQU    $18
56          0019    HVH    EQU    $19
57          001A    HVL    EQU    $1A
58          001B    S1RS2  EQU    $1B
59          001C    ADTRIG EQU    $1C
60          001D    ADTYPE EQU    $1D
61          0020    S1RCTR EQU    $20
62                  *
63          4002    CBITS  EQU    $4002
64          2001    H7134  EQU    $2001
65          2002    L7134  EQU    $2002
66          2003    X7134  EQU    $2003
67          4001    CRA    EQU    $4001
1    **** PPU               ****
0
68          4003    CRB    EQU    $4003
69          4000    DDRA   EQU    $4000
70          4002    DDRB   EQU    $4002
71          003F    PCH    EQU    $3F
72          0040    PCL    EQU    $40
73                  *
74                  * INITIALIZE
75                  *
76  0800 86 02      BEGIN  LDA A  #$02
77  0802 97 05             STA A  SWDEL
78  0804 86 01             LDA A  #$01
79  0806 97 08             STA A  S1DURL
80  0808 97 0A             STA A  S1S2IL
81  080A 97 0C             STA A  S2DURL
82  080C 97 0E             STA A  RHDURL
83  080E 97 0F             STA A  RCYCH
84  0810 97 20             STA A  S1RCTR
85  0812 97 1D             STA A  ADTYPE
86  0814 97 1B             STA A  S1RS2
87  0816 86 F4             LDA A  #$F4
88  0818 97 10             STA A  RCYCL
89  081A 7F 0009           CLR    S1S2IH
90  081D 7F 000D           CLR    RHDURH
91  0820 7F 0007           CLR    S1DURH
92  0823 7F 000B           CLR    S2DURH
93                  *
94  0826 86 0C             LDA A  #$0C
```

```
 95 0828 97 19          STA A  HVH
 96 082A 7F 0011        CLR    P1VH
 97 082D 7F 0017        CLR    RHVH
 98 0830 7F 0015        CLR    P2VH
 99 0833 7F 001A        CLR    HVL
100 0836 7F 0012        CLR    P1VL
101 0839 7F 0018        CLR    RHVL
102 083C 7F 0016        CLR    P2VL
103                *
104 083F 7F 4001        CLR    CRA
105 0842 7F 4003        CLR    CRB
106 0845 7F 4000        CLR    DDRA
107 0848 86 7F          LDA A  #$7F
108 084A B7 4002        STA A  DDRB
109 084D 86 07          LDA A  #$07
110 084F B7 4001        STA A  CRA
111 0852 B7 4003        STA A  CRB
112                *
113 0855 8E 0040        LDS    #$40
114 0858 7F 0000        CLR    PRGSEL
115                *
116 085B 86 08          LDA A  #$08
117 085D 97 02          STA A  SWITRIG
118 085F 86 04          LDA A  #$04
119 0861 97 03          STA A  S1TRIG
1    **** PPU           ****
0
120 0863 86 20          LDA A  #$20
121 0865 97 1C          STA A  ADTRIG
122 0867 7F 0004        CLR    S2TRIG
123 086A 7E 0880        JMP    PROGS
124                *
125                * PROGRAM SELECTOR
126                *
127      0880           ORG    $0880
128 0880 96 00   PROGS  LDA A  PRGSEL
129 0882 81 01          CMP A  #$01
130 0884 27 03          BEQ    VCLAMP
131 0886 7E 0940        JMP    TAP
132                *
133                * VOLTAGE CLAMP SEGMENT
134                *
135                * A/D BOARD TRIGGER
136                *
137 0889 96 1C   VCLAMP LDA A  ADTRIG
138 088B B7 4002        STA A  CBITS
139 088E 86 00          LDA A  #$0
140 0890 B7 4002        STA A  CBITS
141                *
142                * SCOPE TRIGGER
143                *
144 0893 96 01          LDA A  SCTRIG
```

```
145 0895 B7 4002          STA A  CBITS
146 0898 86 00            LDA A  #$0
147 089A B7 4002          STA A  CBITS
148                 *
149                 * P1 DELAY
150                 *
151 089D 7D 0006          TST    P1DEL
152 08A0 27 08            BEQ    P1
153 08A2 96 06            LDA A  P1DEL
154 08A4 BD 09D5 P1D      JSR    D1MS
155 08A7 4A               DEC A
156 08A8 22 FA            BHI    P1D
157                 *
158                 * SET P1 VOLTAGE
159                 *
160 08AA CE 0000 P1       LDX    #$0000
161 08AD 9C 07            CPX    P1DURH
162 08AF 27 15            BEQ    P1P2
163 08B1 96 11            LDA A  P1VH
164 08B3 B7 2001          STA A  H7134
165 08B6 96 12            LDA A  P1VL
166 08B8 B7 2002          STA A  L7134
167 08BB B7 2003          STA A  X7134
168 08BE BD 09D5 P1D2     JSR    D1MS
169 08C1 08               INX
170 08C2 9C 07            CPX    P1DURH
171 08C4 26 F8            BNE    P1D2
1     **** PPU            ****
0
172                 *
173                 * SET P1-P2 VOLTAGE
174                 *
175 08C6 CE 0000 P1P2     LDX    #$0000
176 08C9 9C 09            CPX    P1P2IH
177 08CB 27 15            BEQ    P2
178 08CD 96 13            LDA A  P12VH
179 08CF B7 2001          STA A  H7134
180 08D2 96 14            LDA A  P12VL
181 08D4 B7 2002          STA A  L7134
182 08D7 B7 2003          STA A  X7134
183 08DA BD 09D5 P1P2D    JSR    D1MS
184 08DD 08               INX
185 08DE 9C 09            CPX    P1P2IH
186 08E0 26 F8            BNE    P1P2D
187                 *
188                 * SET P2 VOLTAGE
189                 *
190 08E2 CE 0000 P2       LDX    #$0000
191 08E5 9C 0B            CPX    P2DURH
192 08E7 27 15            BEQ    RH
193 08E9 96 15            LDA A  P2VH
194 08EB B7 2001          STA A  H7134
```

```
   195 08EE 96 16          LDA A  P2VL
   196 08F0 B7 2002        STA A  L7134
   197 08F3 B7 2003        STA A  X7134
   198 08F6 BD 09D5 P2D    JSR    D1MS
   199 08F9 08             INX
   200 08FA 9C 0B          CPX    P2DURH
   201 08FC 26 F8          BNE    P2D
   202                *
   203                * SET RETURN HOLD VOLTAGE
   204                *
   205 08FE CE 0000 RH     LDX    #$0000
   206 0901 9C 0D          CPX    RHDURH
   207 0903 27 15          BEQ    HLD
   208 0905 96 17          LDA A  RHVH
   209 0907 B7 2001        STA A  H7134
   210 090A 96 18          LDA A  RHVL
   211 090C B7 2002        STA A  L7134
   212 090F B7 2003        STA A  X7134
   213 0912 BD 09D5 RHD    JSR    D1MS
   214 0915 08             INX
   215 0916 9C 0D          CPX    RHDURH
   216 0918 26 F8          BNE    RHD
   217                *
   218                * SET HOLD VOLTAGE
   219                *
   220 091A CE 0000 HLD    LDX    #$0000
   221 091D 9C 0F          CPX    RCYCH
   222 091F 27 0D          BEQ    CYCLE
   223 0921 96 19          LDA A  HVH
1    **** PPU               ****
0
   224 0923 B7 2001        STA A  H7134
   225 0926 96 1A          LDA A  HVL
   226 0928 B7 2002        STA A  L7134
   227 092B B7 2003        STA A  X7134
   228                *
   229                * DELAY FOR REMAINDER OF CYCLE
   230                *
   231 092E CE 0000 CYCLE  LDX    #$0000
   232 0931 9C 0F          CPX    RCYCH
   233 0933 27 08          BEQ    ENDIT
   234 0935 BD 09D5 DCYC   JSR    D1MS
   235 0938 08             INX
   236 0939 9C 0F          CPX    RCYCH
   237 093B 26 F8          BNE    DCYC
   238 093D 7E 0880 ENDIT  JMP    PROGS
   239                *
   240                * TAP SEGMENT
   241                *
   242                * A/D BOARD TRIGGER
   243                *
   244 0940 96 1D   TAP    LDA A  ADTYPE
```

```
   245 0942 27 16          BEQ    SCOPE
   246 0944 96 1D          LDA A  ADTYPE
   247 0946 81 01          CMP A  #$1
   248 0948 27 06          BEQ    ADT1
   249 094A 96 20          LDA A  S1RCTR
   250 094C 81 01          CMP A  #$1
   251 094E 26 0A          BNE    SCOPE
   252 0950 96 1C   ADT1   LDA A  ADTRIG
   253 0952 B7 4002        STA A  CBITS
   254 0955 86 00          LDA A  #$0
   255 0957 B7 4002        STA A  CBITS
   256              *
   257              * SCOPE TRIGGER
   258              *
   259 095A 96 01   SCOPE  LDA A  SCTRIG
   260 095C B7 4002        STA A  CBITS
   261 095F 86 00          LDA A  #$0
   262 0961 B7 4002        STA A  CBITS
   263              *
   264              * SAWTOOTH DELAY
   265              *
   266 0964 7D 0005        TST    SWDEL
   267 0967 27 08          BEQ    SW
   268 0969 96 05          LDA A  SWDEL
   269 096B BD 09D5 SWD    JSR    D1MS
   270 096E 4A             DEC A
   271 096F 22 FA          BHI    SWD
   272              *
   273              * SAWTOOTH TRIGGER
   274              *
   275 0971 96 02   SW     LDA A  SWTRIG
1    **** PPU               ****
0
   276 0973 B7 4002        STA A  CBITS
   277 0976 7F 4002        CLR    CBITS
   278              *
   279              * S1 DELAY
   280              *
   281 0979 7D 0006        TST    S1DEL
   282 097C 27 08          BEQ    S1
   283 097E 96 06          LDA A  S1DEL
   284 0980 BD 09D5 S1DL   JSR    D1MS
   285 0983 4A             DEC A
   286 0984 26 FA          BNE    S1DL
   287              *
   288              * SET/HOLD/RESET S1
   289              *
   290 0986 CE 0000 S1     LDX    #$0000
   291 0989 9C 07          CPX    S1DURH
   292 098B 27 10          BEQ    S1S2
   293 098D 96 03          LDA A  S1TRIG
   294 098F B7 4002        STA A  CBITS
```

```
295 0992 BD 09D5 S1D    JSR    D1MS
296 0995 08             INX
297 0996 9C 07          CPX    S1DURH
298 0998 26 F8          BNE    S1D
299 099A 7F 4002        CLR    CBITS
300              *
301              * S1-S2 INTERVAL
302              *
303 099D 7A 0020 S1S2   DEC    S1RCTR
304 09A0 27 03          BEQ    S1S2A
305 09A2 7E 092E        JMP    CYCLE
306 09A5 96 1B   S1S2A  LDA A  S1RS2
307 09A7 97 20          STA A  S1RCTR
308 09A9 CE 0000        LDX    #$0000
309 09AC 9C 09          CPX    S1S2IH
310 09AE 27 08          BEQ    S2
311 09B0 BD 09D5 S1S2D  JSR    D1MS
312 09B3 08             INX
313 09B4 9C 09          CPX    S1S2IH
314 09B6 26 F8          BNE    S1S2D
315              *
316              * SET/HOLD/RESET S2
317              *
318 09B8 CE 0000 S2     LDX    #$0000
319 09BB 9C 0B          CPX    S2DURH
320 09BD 26 03          BNE    S2A
321 09BF 7E 092E        JMP    CYCLE
322 09C2 96 04   S2A    LDA A  S2TRIG
323 09C4 B7 4002        STA A  CBITS
324 09C7 BD 09D5 S2D    JSR    D1MS
325 09CA 08             INX
326 09CB 9C 0B          CPX    S2DURH
327 09CD 26 F8          BNE    S2D
1   **** PPU            ****
0
328 09CF 7F 4002        CLR    CBITS
329              *
330              * JUMP TO CYCLE COUNTER
331              *
332 09D2 7E 092E        JMP    CYCLE
333              *
334              * 1MS DELAY SUBROUTINE
335              *
336 09D5 C6 A3   D1MS   LDA B  #$A3
337 09D7 5A      D1MSB  DEC B
338 09D8 26 FD          BNE    D1MSB
339 09DA 39             RTS
340              *
341              * INTERRUPT HANDLER
342              *
343      0B00           ORG    $0B00
344 0B00 CE 0000        LDX    #$0000
```

```
345 0B03 C6 1E          LDA B  #$1E
346 0B05 BD 0B1C INT1   JSR    DATGET
347 0B08 08             INX
348 0B09 5A             DEC B
349 0B0A 22 F9          BHI    INT1
350 0B0C 96 1B          LDA A  S1RS2
351 0B0E 97 20          STA A  S1RCTR
352 0B10 8E 0039        LDS    #$39
353 0B13 86 08          LDA A  #$08
354 0B15 97 3F          STA A  PCH
355 0B17 86 80          LDA A  #$80
356 0B19 97 40          STA A  PCL
357 0B1B 3B             RTI
358             *
359             * DATA HANDLING SUBROUTINE
360             *
361 0B1C B6 4002 DATGET LDA A  CBITS
362 0B1F 2A FB          BPL    DATGET
363 0B21 B6 4000        LDA A  DDRA
364 0B24 A7 00          STA A  0,X
365 0B26 B6 4002 B7     LDA A  CBITS
366 0B29 2B FB          BMI    B7
367 0B2B 39             RTS
368             *
369                     END


                NO ERRORS
```